

AD-A094 766

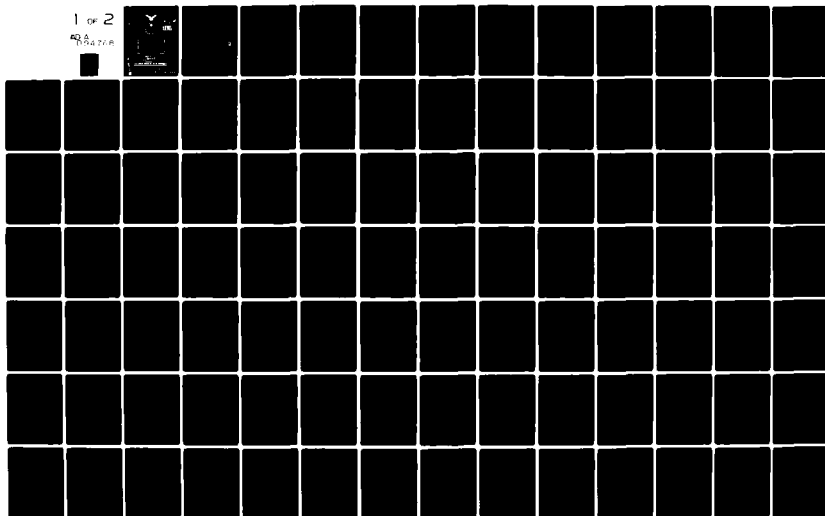
AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 12/2
MODERN OPTIMAL CONTROL METHODS APPLIED IN ACTIVE CONTROL OF A T--ETC(U)
DEC 80 A M JANISZEWSKI
AFIT/65/AA/80D-2

UNCLASSIFIED

NL

1 OF 2

Q2
THRU 200



AD A094766



LEVEL II



14

AFIT/GA/AA/80D-2

23 JAN 1981

APPROVED FOR PUBLIC RELEASE AFR 190-17.

Laurel A. Lampela

LAUREL A. LAMPELA, 2Lt, USAF
Deputy Director, Public Affairs

Air Force Institute of Technology (ATC)
Wright-Patterson AFB, OH 45433

12 109

9 Master's thesis

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/ _____	
Availability Codes	
Dist	Avail and/or Special
A	

6 MODERN OPTIMAL CONTROL METHODS
APPLIED IN ACTIVE CONTROL OF A
TETRAHEDRON.

DTIC
ELECTE
FEB 10 1981
F

THESIS

AFIT/GA/AA/80D-2 Alan M. Janiszewski
10 captain USAF

Michael

11 Dec 80

Approved for public release; distribution unlimited

012225

Lee

MODERN OPTIMAL CONTROL METHODS
APPLIED IN ACTIVE CONTROL OF A
TETRAHEDRON

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirement for the Degree of
Master of Science

by
Alan M. Janiszewski
Captain USAF
Graduate Astronautical Engineering
December 1980

Preface

I would like to express gratitude to my thesis advisor, Dr. R. A. Calico, for providing an underlying sense of direction to this study, while willingly spending his time guiding me through the concepts contained herein. Additionally, the support of Capt. J. Rader and Capt. W. Wiesel through their sequences in optimization techniques and estimation theory clearly enhanced my understanding of concepts germane to this analysis. Also, Capt. J. Silverthorn provided valuable insight into more clearly presenting many of the ideas which follow. I'd like to thank my wife and typist, Grace. Her understanding and support in the former role meant far more than her assistance in the latter. Finally, I am indebted to my son, Andy, who only knew I wasn't there without being able to understand why.

Alan M. Janiszewski

Contents

Preface	ii
List of Figures	iv
List of Tables	v
Abstract	vi
I. Introduction	1
II. System Model	5
General Configuration	5
Equations of Motion	10
Modal Control	17
Block Diagram Representation for the Linear Model	22
III. Transformation Matrix Control Method	25
IV. The Computer Model	31
V. Investigation	35
Outline	35
Elimination of Control Spillover	36
Elimination of Observation Spillover	49
Sensor Additions	56
VI. Conclusions	63
VII. Recommendations	65
Bibliography	66
Appendix A	67
Appendix B	70
Vita	98

List of Figures

Page

1. Cross Sectional View of the System Model	6
2. View of System Model Down Y and Z Axes	7
3. Simple Open Loop Plant	17
4. System in Block Diagram Form	23
5. System in Modified Block Diagram Form	23
6. LOSX VS. Time, $B_s G \neq 0$, $F = 1.0$	40
7. LOSY VS. Time, $B_s G \neq 0$, $F = 1.0$	41
8. LOSX VS. Time, $B_s G = 0$, $F = 1.0$	42
9. LOSY VS. Time, $B_s G = 0$, $F = 1.0$	43
10. LOSX VS. Time, $B_s G \neq 0$, $F = 1000.0$	45
11. LOSY VS. Time, $B_s G \neq 0$, $F = 1000.0$	46
12. LOSX VS. Time, $B_s G = 0$, $F = 1000.0$	47
13. LOSY VS. Time, $B_s G = 0$, $F = 1000.0$	48
14. LOSX VS. Time, $KC_s \neq 0$, $Q = 1000.0$	52
15. LOSY VS. Time, $KC_s \neq 0$, $Q = 1000.0$	53
16. LOSX VS. Time, $KC_s \neq 0$, $Q = 1000.0$	54
17. LOSY VS. Time, $KC_s \neq 0$, $Q = 1000.0$	55
18. LOSX VS. Time, $KC_s \neq 0$, 8 Sensors, $Q = 1000.0$	59
19. LOSY VS. Time, $KC_s \neq 0$, 8 Sensors, $Q = 1000.0$	60
20. LOSX VS. Time, $KC_s = 0$, 8 Sensors, $Q = 1000.0$	61
21. LOSY VS. Time, $KC_s = 0$, 8 Sensors, $Q = 1000.0$	62

List of Tables

I. Node Coordinates	8
II. Key Results of NASTRAN Eigenvalue Analysis	9
III. Initial Conditions for Time History Response	9
IV. Elimination of Control Spillover; $F = 1.0$	39
V. Elimination of Control Spillover; $F = 1000.0$	44
VI. Elimination of Observation Spillover; $Q = 1000.0$	50
VII. Singular Values of C_s ; Six Non-reoriented Sensors.	49
VIII. Reduction of Observation Spillover; $Q = 1000.0$ (Fixed Sensors)	51
IX. Elimination of Observation Spillover; $Q = 1000.0$ (8 Sensors)	58

Abstract

Modern optimal control methods are applied to a lumped mass model of a tetrahedron. The four unit masses of this model are interconnected by isotropic massless rods which are capable of axial deformation only (no bending). NASTRAN is employed in generating a normal modes approximation, while providing the mode shapes and frequencies for the resultant twelve modes. System control is achieved via collocated sensor/actuator pairs at three of the four masses. Pointing accuracy at the fourth mass is used as a figure of merit in determining the effectiveness of the controller. A prescribed line of sight response is established as a goal for successful control.

The controller is developed using linear optimal techniques which produce feedback gains proportional to the state. The state is represented as modal amplitudes and velocities as determined by the sensors. The four higher frequency modes are truncated to signify a simplifying order reduction step. State estimation is incorporated due to the non-availability of modal amplitudes and velocities. The feedback gains are established via steady state optimal regulator theory. Control is applied with point force actuators. System response is examined in light of the effects of observation spillover and control spillover onto a specified number of suppressed modes. A comparison is obtained by complete elimination of the spillover effect. Using singular value decomposition, the spillover is first eliminated through judicious reorientation of one sensor/actuator pair. An attempt to control two modes and suppress six demonstrates the

advantages of spillover elimination, but fails to satisfy the specified response criteria.

Sensors are added to the model at the fourth mass and observation spillover is again eliminated. Line of sight response was improved over the case without sensor additions, but was still inadequate. The truncated modes were added to the system with little degradation, verifying the acceptability of this truncation.

MODERN OPTIMAL CONTROL METHODS
APPLIED IN ACTIVE CONTROL OF A
TETRAHEDRON

Introduction

The potential for larger and more complex space structures has grown concurrent with the approach of an active, operable space transportation program. Present system concepts involve the deployment of earth resource satellites and micro-wave power relay systems with dimensions extending to hundreds and eventually thousands of meters in diameter. A key design criteria for these immense, mechanically flexible systems is the requirement to develop advanced methods for control. More precisely, a principle issue in the control of a system with an infinite number of vibrational modes is the generation of a method for stabilizing these huge structures with dimensionally realistic controllers. This requirement is basically a function of on-board computer, sensor, and force actuator limitations, along with incumbent modelling inaccuracies. Of the numerous methods now being examined as potential solutions to this control problem, modern state space control theory has received general acceptance as the most viable technique. Applying classical control methods to these large structures is seen as computationally improbable; at the same time, modern state space theory, incorporating a finite element system representation, can be successfully applied to a very wide

class of flexible structures. This theory is most commonly applied using an optimal time-invariant linear regulator as a means of actively controlling vibration.

Due to the inherent hardware limitations briefly highlighted, active control must be restricted to a relatively small number of critical modes. Therefore, in a necessary truncation step, some higher frequency modes remain unmodelled. Natural damping in the system is assumed to preclude the possibility of instability resulting from these modes. Of the remaining modes (still a potentially large number) it is further desirable to treat only a critical few (not necessarily those with the lowest frequency), while suppressing the rest.

However, the sensor outputs are contaminated by the remaining "suppressed" modes, and the eventual feedback control also excites these modes. Balas (Ref 1) labels these effects "observation spillover" and "control spillover" respectively. He shows that either or both of these effects can lead to overall instabilities; the suppressed modes must, as a result, be a design consideration. Balas describes a technique with which to develop a feedback controller using state variable methods. The key to this approach is the use of narrow bandpass filters which effectively comb out the suppressed mode frequencies to eliminate observation spillover.

Another method for developing an appropriate feedback controller was first presented by Sesak (Ref 2), and later expanded by Coradetti (Ref 3). This approach involves the use of a so-called "singular perturbation" technique in analyzing and eliminating the spillover-generated instabilities. Coradetti concludes

that employing this "singular perturbation" method in a limiting sense, with an infinite penalty applied against any spillover, is equivalent to finding a transformation matrix. This transformation matrix, when applied to feedback gains, effectively eliminates any spillover terms. It should be noted that, even if spillover does not render the system unstable, applying the transformation method may still improve performance. Additionally, while no method for actually automating optimal sensor and actuator placement is defined, some valuable insight into the nature of this task is precipitated. This is accomplished utilizing what have now become well known state space control techniques in conjunction with singular value decomposition of the rectangular matrices of modal amplitudes (Ref 4).

The principle function of this thesis is to provide application of the Coradetti approach to a three dimensional, lumped mass model of a tetrahedron. A line of sight at one of the masses (simulating pointing accuracy) will be used as a figure of merit with which to judge the general effectiveness of this method. This thesis will serve as a direct extension of the work done by Sanborn (Ref 5), in which the stability of a cantilever beam in bending vibration was studied. Specifically, this thesis will examine model response as affected by the number and orientation of position sensors and force actuators. The elimination of control spillover and observation spillover will be obtained using singular perturbation and singular value decomposition techniques.

A representation of a tetrahedron has been obtained via the normal modes approximation package found in the NASTRAN finite

element computer program. The natural frequencies and eigenvalues/eigenvectors associated with each mode were provided by a study done by the Charles Stark Draper Laboratory. For application of the control method, position sensors are used to evaluate modal amplitudes, while point force actuators accomplish the state variable feedback control. Singular value decomposition of the matrices of modal amplitudes at sensor locations and actuator locations is used to produce a transformation matrix by which spillover terms are eliminated. A model with higher order modes truncated (un-modelled) is used to design the controller. The effectiveness of this controller against all modes is examined. Finally, a study of improved performance with added sensors is generated.

System Model

General Configuration

Of the many design criteria which must be considered for the large flexible spacecraft currently being advanced, pointing accuracy looms as the most critical. As a function of system size and operating frequency, pointing accuracies in the range of one tenth the half-power signal beam width will be required. The ability to meet these stringent requirements becomes a direct function of the isotropic stiffness of the system. One of the space erectable or assembly concepts that has the promise of supplying this needed stiffness in larger systems is the geodetic truss (Ref 6). Based on current Space Shuttle cargo capacities, whole units of up to 91.4 meters can be packaged for deployment. For very large systems, these units are assembled as an amalgamation of tetrahedrons-- the basic unit of geodetic truss. By changing the size of the tetrahedrons, a large array of varying stiffness antenna substructures can be developed. For this reason, a tetrahedron is seen as an important model against which to apply proposed control techniques.

The finite element, lumped mass model to be used herein is depicted in Fig 1 and Fig 2. This model is seen to consist of ten nodes. The twelve interconnecting truss members are assumed to be massless and are capable of resisting or exerting axial force only (no-bending). Masses are of one unit each, and are located at grid points one through four. Each mass is capable of perturbation with three degrees of freedom.

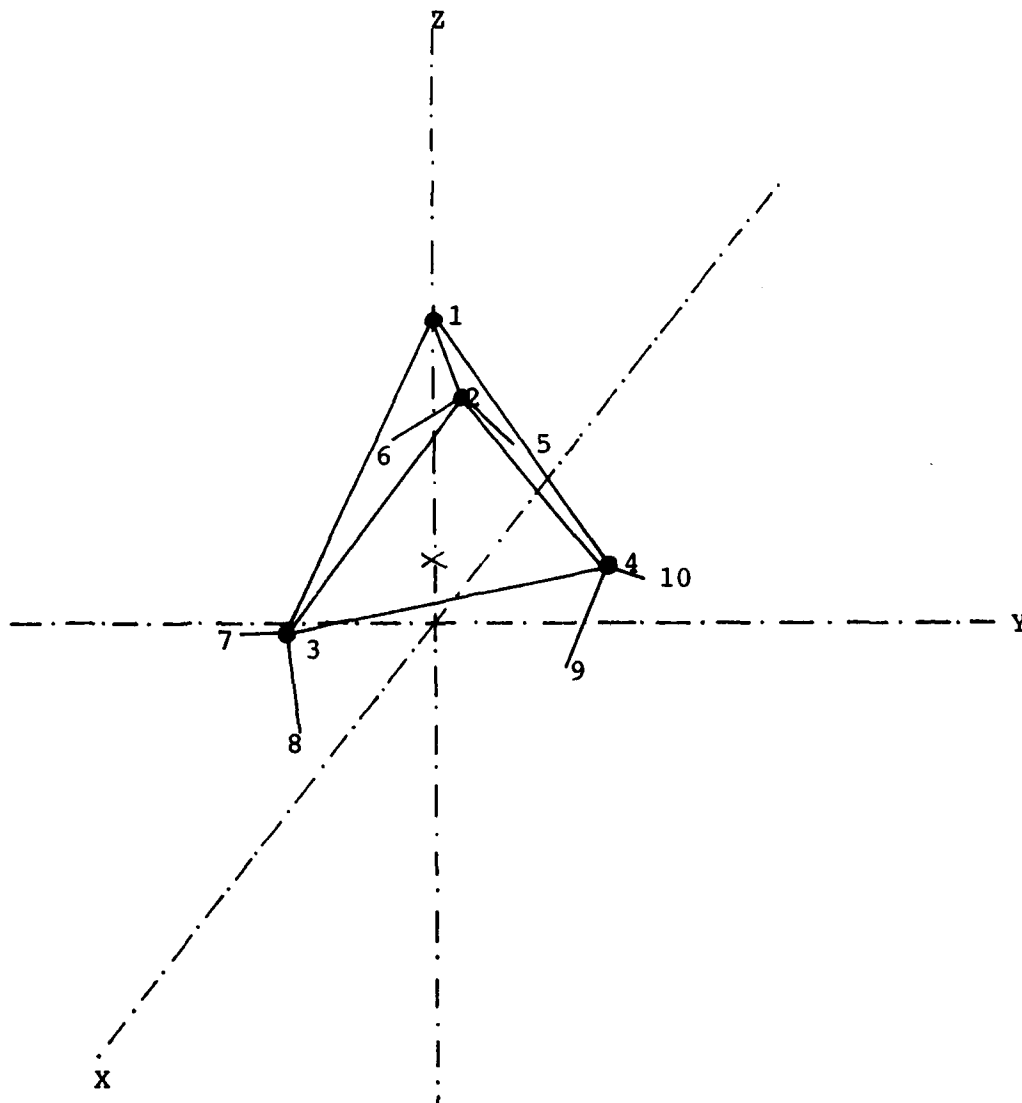


Figure 1. Cross Sectional View of the System Model

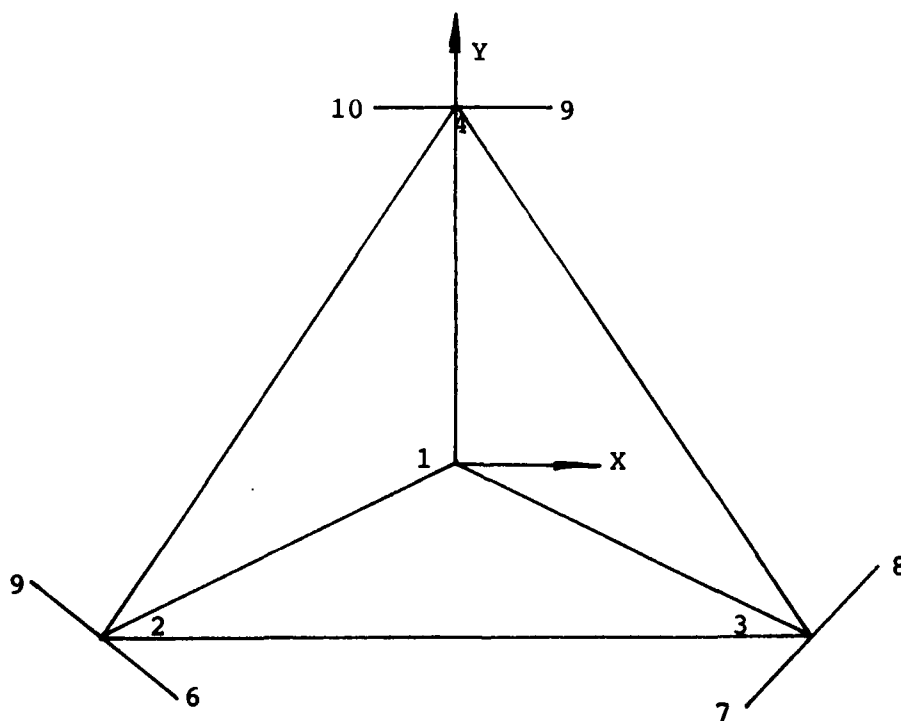
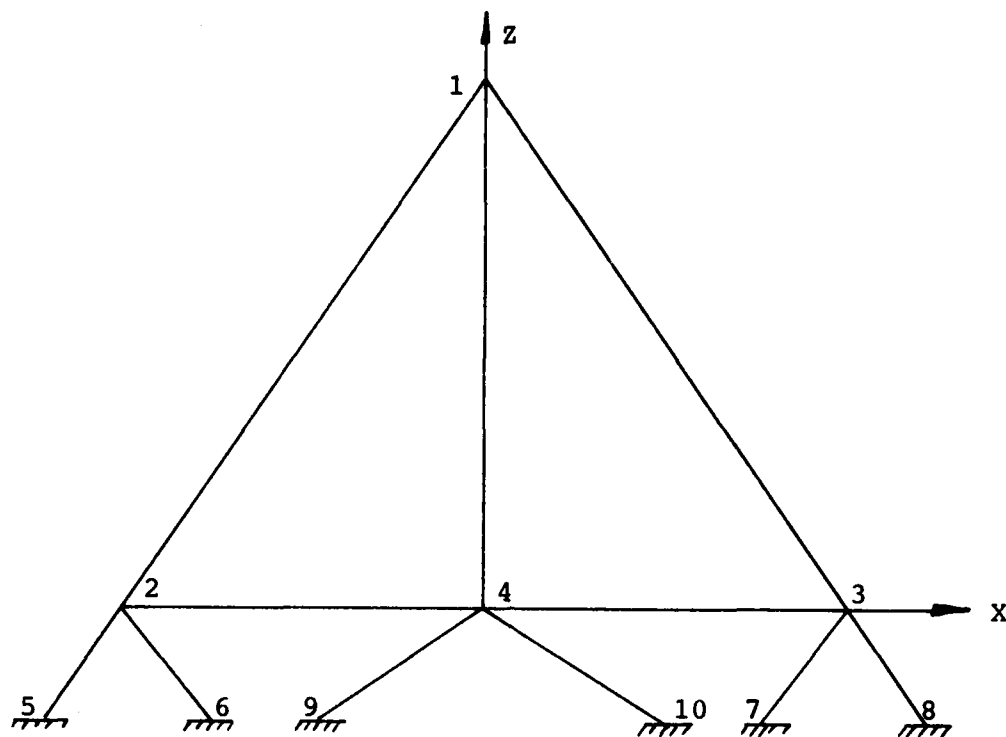


Figure 2. View of System Model Down Y and X axes

The remaining grid points (five through ten) serve to establish a fixed line of sight for an initial set of six collocated sensor/actuator pairs. Node coordinates for the model are listed as Table I. For this analysis, position sensors are employed, but velocity sensors are not. The effects of this are detailed in the linear system model to be developed in this section.

An eigenvalue analysis of this nominal model has been provided via the NASTRAN computer program. Key results of this analysis are presented as Table II. The eigenvectors associated with these eigenvalues can be found in Appendix A. Table III provides the initial conditions required for a time history examination of system stability. This stability will be assessed using, as a figure of merit, the pointing accuracy along the Z axis at node 1. Since any perturbation directly along the Z axis has no impact on pointing accuracy, the line of sight in the X and Y directions only will be examined.

Table I.

Node Coordinates

<u>Node</u>	<u>X</u>	<u>Y</u>	<u>Z</u>
1	0.0	0.0	10.165
2	-5.0	-2.887	2.0
3	5.0	-2.887	2.0
4	0.0	5.7735	2.0
5	-6.0	-1.1547	0.0
6	-4.0	-4.6188	0.0
7	4.0	-4.6188	0.0
8	6.0	-1.1547	0.0
9	2.0	5.7735	0.0
10	-2.0	5.7735	0.0

Table II

Key Results of NASTRAN Eigenvalue Analysis

<u>Mode</u>	<u>Generalized Mass</u>	<u>Generalized Stiffness</u>	<u>ω_n (rad)/sec</u>	<u>Ω (rad²)/sec²</u>
1	1.0E+00	1.37E+00	1.17E+00	1.37E+00
2	1.0E+00	2.15E+00	1.47E+00	2.15E+00
3	1.0E+00	8.79E+00	2.96E+00	8.79E+00
4	1.0E+00	1.26E+01	3.56E+00	1.26E+01
5	1.0E+00	1.48E+01	3.85E+00	1.48E+01
6	1.0E+00	2.65E+01	5.15E+00	2.65E+01
7	1.0E+00	3.22E+01	5.67E+00	3.22E+01
8	1.0E+00	3.26E+01	5.71E+00	3.26E+01
9	1.0E+00	7.99E+01	8.93E+00	7.99E+01
10	1.0E+00	1.06E+02	1.03E+01	1.06E+02
11	1.0E+00	1.19E+02	1.09E+01	1.19E+02
12	1.0E+00	1.95E+02	1.40E+01	1.95E+02

Table III

Initial Conditions for Time History Response

<u>Mode</u>	<u>Displacement (η)</u>	<u>Velocity ($\dot{\eta}$)</u>
1	-.001	-.003
2	.006	.01
3	.001	.03
4	-.009	-.02
5	.008	.02
6	-.001	-.02
7	-.002	-.003
8	.002	.004
9	.0	.0
10	.0	.0
11	.0	.0
12	.0	.0

Equations Of Motion

Since there are no exact equations of motion for a continuous model of a tetrahedron, we are restricted to the discretized representation provided by the finite element routines. The output function or motion of the model can be expressed as:

$$Y(x_j, t) = \sum_{i=1}^n \phi_i(x_j) U_i(t) \quad (1)$$

where the $\phi_i(x_j)$ terms are the mode shapes, and the $U_i(t)$ terms are the mode amplitudes, with n being the number of modes exhibited by the model. For an exact solution to a continuous system, the number of lumped masses and the number of modes (n) would have to reach infinity. Practically speaking, the total system displacement $Y(x_j, t)$, can be reasonably represented by a truncation of Eq (1) such that n is considerably less than infinity. This truncation will, of course, lead to model reduction errors; but, up to a certain point these errors are relatively insignificant.

NASTRAN analyzes the model in Fig 1. and generates both the normal mode shapes and the corresponding natural frequencies (ω_n). Since this is a lumped mass model consisting of four masses, with each mass having three degrees of freedom, there are a total of twelve normal modes.

Linear System Model

As stated, the number of modes (n) for a complex model may be very large. The practical limitations for an on-board computer and the associated sensor and actuator hardware make it

necessary to develop a controller that is concerned with a minimum number of modes, while still satisfying what may be very stringent requirements on the performance (here, line of sight accuracy). As the control theory outlined in this paper is elaborated, a possible method for determining which modes require control will be discussed. At this point, assuming this determination is possible, the system output of Eq (1) can be segregated into 3 partitions; the controlled, the suppressed, and the unmodelled:

$$Y(x_j, t) = Y_c(x_j, t) + Y_s(x_j, t) + Y_{um}(x_j, t) \quad (2)$$

$Y_{um}(x_j, t)$ is that portion of the output generated through the highest frequency modes. These modes are unmodelled, with the hope that the bandwidths of the sensors and actuators employed will be less than the natural frequencies of the modes. Furthermore, since these modes have such high frequencies, they may be quite difficult to excite. Hence, any controller designed for this system can ignore these modes. These modes are subsequently called the residual modes.

$Y_s(x_j, t)$ is that portion of the output generated by modes of less high frequency, which, none-the-less have a minimal direct impact on system performance. Due to their indirect and potentially destabilizing impact (spillover), they must be included in the design process. These modes are subsequently called the suppressed modes.

$Y_c(x_j, t)$ is that portion of the output which we must directly control to insure satisfactory performance. These critical modes

will subsequently be called the controlled modes.

Equation (2) can now be written in segregated form as:

$$Y_c(x_j, t) = \sum_{i=1}^c \phi_i(x_j) \bar{U}_i(t) \quad (3)$$

$$Y_s(x_j, t) = \sum_{i=c+1}^{c+s} \phi_i(x_j) \bar{U}_i(t) \quad (4)$$

$$Y_{um}(x_j, t) = \sum_{i=c+s+1}^n \phi_i(x_j) \bar{U}_i(t) \quad (5)$$

where c is the number of controlled modes, s is the number of suppressed modes, and n is the total number of modes in the model. Again, for this system model, n is twelve. For the purpose of future analysis, the case of truncating the highest frequency modes will be simulated by suggesting that the last four (highest natural frequency) modes generated by NASTRAN fall into this category. The design process for the overall controller will be based on knowledge of only the first eight modes. The eventual controller will be applied to a system incorporating all twelve modes in an attempt to verify the acceptability of this truncation. The modelling can thus be seen as a process of two truncations in the effort to reduce control hardware and software requirements. First, the model is truncated to a workable number of modes by designing a controller that is blind to the higher frequency modes. Second, the model is limited to controlling only the critical modes where the figure of merit is concerned.

NASTRAN has taken the prescribed system with the masses and gridpoints provided, and modelled the structure with a set

of second order differential equations. These are the basic spring mass differential equations such that $\ddot{\eta} + \omega_n^2 \eta = f$. The associated first order eigenproblem is solved (Ref 7) so as to provide the decoupled normal modes. This allows assembling a state space representation of the system:

$$\dot{\bar{X}}(t) = A\bar{X}(t) + B\bar{u}(t) \quad (6)$$

where

$\bar{X}(n \times 1)$ is the state vector

$\bar{u}(m \times 1)$ is the control input vector

$A(n \times n)$ is the plant matrix

$B(n \times m)$ is the input matrix

By letting the state X be the partitioned matrix of mode amplitudes ($\bar{U}_i(t)$) and their rates of change ($\dot{\bar{U}}_i(t)$) the state variables become:

$$\bar{X}(t) = \left[\bar{U}_i(t) \quad \vdots \quad \dot{\bar{U}}_i(t) \right]^T \quad i = 1, 2, \dots, n \quad (7)$$

Further separating the states into \bar{X}_c , formed by the controlled amplitudes and rates; and \bar{X}_s , formed by the suppressed amplitudes and rates renders:

$$\bar{X}_c(t) = \left[\bar{U}_i(t) \quad \vdots \quad \dot{\bar{U}}_i(t) \right]^T \quad i = 1, 2, \dots, c \quad (8)$$

$$\bar{X}_s(t) = \left[\bar{U}_j(t) \quad \vdots \quad \dot{\bar{U}}_j(t) \right]^T \quad j = c+1, \dots, c+s \quad (9)$$

Substituting these states into Eq (6), the system is now modelled by:

$$\dot{\bar{X}}_c(t) = A_c \bar{X}_c(t) + B_c \bar{u}(t) \quad (10)$$

$$\dot{\bar{X}}_s(t) = A_s \bar{X}_s(t) + B_s \bar{u}(t) \quad (11)$$

The system parameter matrices are defined as:

$$A_c = \begin{bmatrix} 0 & \vdots & I \\ \dots & & \dots \\ -\bar{\Omega}_c & \vdots & -2\xi_i \bar{\omega}_i \end{bmatrix} \quad (12)$$

$$A_s = \begin{bmatrix} 0 & \vdots & I \\ \dots & & \dots \\ -\bar{\Omega}_s & \vdots & -2\xi_j \bar{\omega}_j \end{bmatrix} \quad (13)$$

$$B_c = \begin{bmatrix} 0 \\ \dots \\ B_{\sim c} \end{bmatrix} \quad (14)$$

$$B_s = \begin{bmatrix} 0 \\ \dots \\ B_{\sim s} \end{bmatrix} \quad (15)$$

The $\bar{\omega}_i$ and $\bar{\omega}_j$ terms are the diagonal elements of square matrices which represent the natural frequencies of the controlled and suppressed modes respectively, while the ξ_i and ξ_j terms represent the damping ratios for those modes $i = 1, 2, \dots, c$ and $j = c+1, c+2, \dots, c+s$. The $\bar{\Omega}_c$ and $\bar{\Omega}_s$ terms are diagonal matrices of these same natural frequencies squared as determined by NASTRAN. Therefore, as an example, if two modes for a given system were to be controlled, one would have:

$$A_c = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\omega_1^2 & 0 & -2\xi_1\omega_1 & 0 \\ 0 & -\omega_2^2 & 0 & -2\xi_2\omega_2 \end{bmatrix} \quad (16)$$

Furthermore, the B_c and B_s matrices are the control input matrices, and are those matrices whose columns are the mode shapes $(\phi_i(x), \phi_j(x))$ evaluated at each actuator location such that:

$$B_c = \begin{bmatrix} \phi_1(x_1) & \phi_1(x_2) & \dots & \phi_1(x_a) \\ \phi_2(x_1) & \phi_2(x_2) & \dots & \phi_2(x_a) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_c(x_1) & \phi_c(x_2) & \dots & \phi_c(x_a) \end{bmatrix} \quad (17)$$

$$B_s = \begin{bmatrix} \phi_{c+1}(x_1) & \phi_{c+1}(x_2) & \dots & \phi_{c+1}(x_a) \\ \phi_{c+2}(x_1) & \phi_{c+2}(x_2) & \dots & \phi_{c+2}(x_a) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{c+s}(x_1) & \phi_{c+s}(x_2) & \dots & \phi_{c+s}(x_a) \end{bmatrix} \quad (18)$$

where a is the total number of actuators employed.

Additionally, state space methods render the sensor output as:

$$\bar{Y}(t) = C_c \bar{X}_c(t) + C_s \bar{X}_s(t) \quad (19)$$

with

$$C_c = \begin{bmatrix} C_c & \vdots & 0 \end{bmatrix} \quad (20)$$

$$C_s = \begin{bmatrix} C_c & \vdots & 0 \end{bmatrix} \quad (21)$$

where C_c and C_s are matrices whose "rows" are the mode shapes of the controlled and the suppressed modes respectively evaluated at the prescribed sensor locations such that:

$$C_c = \begin{bmatrix} \phi_1(x_1) & \phi_2(x_1) & \dots & \phi_c(x_1) \\ \phi_1(x_2) & \phi_2(x_2) & \dots & \phi_c(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(x_b) & \phi_2(x_b) & \dots & \phi_c(x_b) \end{bmatrix} \quad (22)$$

$$C_s = \begin{bmatrix} \phi_{c+1}(x_1) & \phi_{c+2}(x_1) & \dots & \phi_{c+s}(x_1) \\ \phi_{c+1}(x_2) & \phi_{c+2}(x_2) & \dots & \phi_{c+s}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{c+1}(x_b) & \phi_{c+2}(x_b) & \dots & \phi_{c+s}(x_b) \end{bmatrix} \quad (23)$$

The null portion of the C_c and C_s matrices represent the velocities at the prescribed sensor locations, which are zero since only displacement sensors are being employed. Again, b is the total number of sensors used. It should be clear that if collocated sensors and actuators are used, with $a = b$, then

$$B_c^T = C_c \quad (24)$$

$$B_s^T = C_s \quad (25)$$

As these model elements are created, it becomes clear that this methodology is independent of structural complexity, except for the overall matrix dimensions. Therefore, the applicability of the subsequent analysis can be seen to be far reaching.

As a starting point toward developing a state variable feedback controller, Fig 3 below represents the uncontrolled system that has been here-to-fore described.

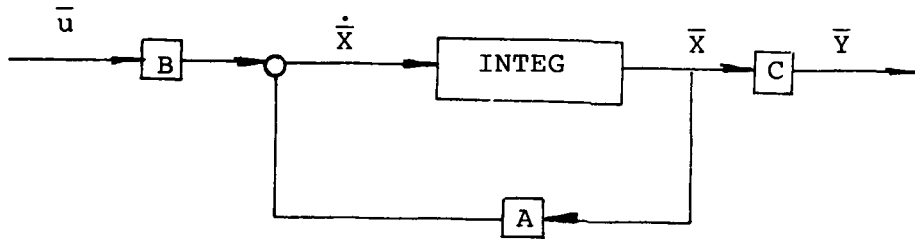


Figure 3. Simple Open Loop Plant

In order to eventually form an active control, $u(t)$, using state variable (modern control) feedback techniques, complete knowledge of the actual state at time t must be known. However, the only measure of the state \bar{X} is the measurement vector \bar{Y} provided by the sensors. To take those observations and create the corresponding state, it will be necessary to develop a state estimator which will accept those sensor observations and estimate \bar{X} as \hat{X} .

Modal Control

As Balas explains, the state estimator used in developing active feedback control can either be a Kalman Filter when it is found that the signal-to-noise ratios are relatively small, or a Luenberger observer, or a least squares technique, should the signal to noise ratio be high enough to treat the system

as deterministic. Regardless of which is used, the estimator will have the form:

$$\dot{\hat{X}}_N(t) = A_N \bar{X}_N + B_N \bar{u}(t) + K_N [\bar{Y}(t) - \hat{Y}(t)] \quad (26)$$

and
$$\hat{X}_N(0) = 0 \quad (27)$$

$$\hat{Y}(t) = C_N \hat{X}_N \quad (28)$$

where N is replaced in our system by either c or s.

Observation of Eq (26) shows that the estimator equation is comprised of the internal model of the state as in Eqs (10) and (11), plus a correction term which is made up of the error between the measured output ($\bar{Y}(t)$) and the computed output ($\hat{Y}(t)$). Equation (27) establishes an initial condition for the state out of convenience. The error in this state estimation process is given as:

$$\bar{e}_N(t) = \bar{X}_N(t) - \hat{X}_N(t) \quad (29)$$

The equations for this estimator error, formed by combining Eqs (26), (27), and (28) with Eqs (10), (11), and (19) becomes:

$$\dot{\bar{e}}_N(t) = (A_N - K_N C_N) \bar{e}_N(t) + K_N C_N \bar{X}_R(t) \quad (30)$$

For the prescribed system, this becomes:

$$\dot{\bar{e}}_c(t) = (A_c - K_c C_c) \bar{e}_c(t) + K_c C_s \bar{X}_s(t) \quad (31)$$

Ignoring the suppressed modes, this finally becomes:

$$\dot{\bar{e}}_c(t) = (A_c - K_c C_c) \bar{e}_c(t) \quad (32)$$

The observer gain matrix, K , must be formulated so as to insure that the estimator error defined in Eq (32) decays exponentially at a rate more rapid than the system dynamics. The decay rate is determined by the eigenvalues of $(A_c - KC_c)$. Since the eigenvalues of a matrix are equal to the eigenvalues of the transpose of that matrix, Eq (32) can be rewritten as follows:

$$\dot{\bar{w}}(t) = A_c^T \bar{w}(t) - C_c^T \bar{g}(t) \quad (33)$$

$$\bar{g}(t) = K^T \bar{w}(t) \quad (34)$$

The observer gain matrix, K , can now be calculated via steady state optimal regulator theory. This is equivalent to minimizing the quadratic regulator performance index J , where:

$$J = \frac{1}{2} \int_0^\infty (\bar{w}^T Q_{ob} \bar{w} + \bar{g}^T R_{ob} \bar{g}) dt \quad (35)$$

The known optimal solution to this minimization problem is:

$$K^T = -R_{ob}^{-1} C_c^T P \quad (36)$$

where P is the solution to the steady state algebraic matrix Ricatti Equation:

$$P A_c + A_c^T P - P C_c^T R_{ob}^{-1} C_c P + Q_{ob} = 0 \quad (37)$$

where Q_{ob} is an $n \times n$ positive semidefinite state weighting matrix

R_{ob} is an $m \times m$ positive definite control weighting matrix

By treating only the controlled modes in the generation

of the optimal state feedback gain matrix, we have significantly reduced the order of the controller. This was accomplished, as previously stated to avoid practical problems encountered in deriving a global controller. The reduced order controller will subsequently be designed to control a subset of all of the system states, while simultaneously avoiding any excitation of the remaining states. Coradetti clarifies the advantages of this process when he points out that the computational burden of solving the Ricatti Equation increases roughly as the cube of the order of the equation. There may simply not be sufficient on-board computer memory available. Also, the state estimator process increases with system order at a greater than linear rate. Finally, with non-interacting controller there will be greater fault tolerance to actuator failures.

In precisely the same fashion, the control feedback gain matrix, G , can be formulated. Now, again using steady state optimal regulator theory, the performance index to be minimized is:

$$J = \frac{1}{2} \int_0^{\infty} (\bar{X}_c^T F \bar{X}_c + \bar{f}^T R \bar{f}) dr \quad (38)$$

where

F is an $n \times n$ positive semidefinite state weighting matrix

R is an $m \times m$ positive definite control weighting matrix

The optimal solution to this minimization problem is

$$G = R^{-1} B_c^T S \quad (39)$$

where g is the solution to the matrix Ricatti Equation:

$$SA_c + A_c^T S - SB_c R^{-1} B_c^T S + F = 0 \quad (40)$$

Implementing the results of Eqs (36) and (39) with the system Eqs (10) and (11), as well as Eq (32) renders:

$$\dot{\bar{X}}_c(t) = (A_c + B_c G) \bar{X}_c(t) + B_c G \bar{e}(t) \quad (41)$$

$$\dot{\bar{X}}_s(t) = A_s \bar{X}_s(t) + B_s G \bar{X}_c(t) + B_s G \bar{e}(t) \quad (42)$$

By taking the step of defining a system state vector incorporating the controlled states, the suppressed states, and the estimator error, such that:

$$\bar{Z}(t) = \begin{bmatrix} \bar{X}_c^T(t) & \vdots & \bar{e}^T(t) & \vdots & \bar{X}_s^T(t) \end{bmatrix}^T \quad (43)$$

a closed loop system model, containing the effects of the suppressed and controlled modes, and utilizing state variable feedback as the control mechanism can be presented as:

$$\dot{\bar{Z}}(t) = \begin{bmatrix} A_c + B_c G & \vdots & B_c G & \vdots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \vdots & A_c - K C_c & \vdots & K C_s \\ \dots & \dots & \dots & \dots & \dots \\ B_s G & \vdots & B_s G & \vdots & A_s \end{bmatrix} \bar{Z}(t) \quad (44)$$

Recalling that the observation and control feedback gain matrices (K and G) were designed to operate on the controlled modes, the terms $K C_s$ and $B_s G$ create potential problems. These in effect, are known as observation spillover and control spillover, respectively. Although all of the diagonal matrices of

Eq (44) are designed to have purely negative real parts for all eigenvalues, it is obvious that the KC_s and $B_s G$ terms can cause overall system instabilities.

Block Diagram Representation for the Linear Model

In a parallel section of his paper, Sanborn generates the block diagrams representing this new system model in two separate forms. Since the equations now governing the model are:

$$\dot{\bar{X}} = A\bar{X} + B\bar{u} \quad \text{State Equation} \quad (45)$$

$$\bar{Y} = C\bar{X} \quad \text{Output Equation} \quad (46)$$

$$\bar{u} = G\hat{X}_C \quad \text{Control Equation} \quad (47)$$

$$\dot{\hat{X}}_C = A_C \hat{X}_C + B_C \bar{u} + K(\bar{Y} - \hat{Y}) \quad \text{Estimator Equation} \quad (48)$$

The system can be presented as Fig 4. This diagram can be manipulated per Johnson (Ref 8) to generate a modified block diagram form as shown in Fig 5.

From Fig 5 the closed loop transfer function for the controller is seen to be:

$$\frac{f}{y}(s) = K(SI - A_C - B_C G + KC_C)^{-1} G \quad (49)$$

From this transfer function, we know that if any of the eigenvalues of $(A_C + B_C G - KC_C)$ are positive, then the controller is unstable. Since the techniques for generating both the observation and control gain matrices were employed independently, the possibility that an unstable controller is formed exists. Although the controller, when coupled with the plant, would

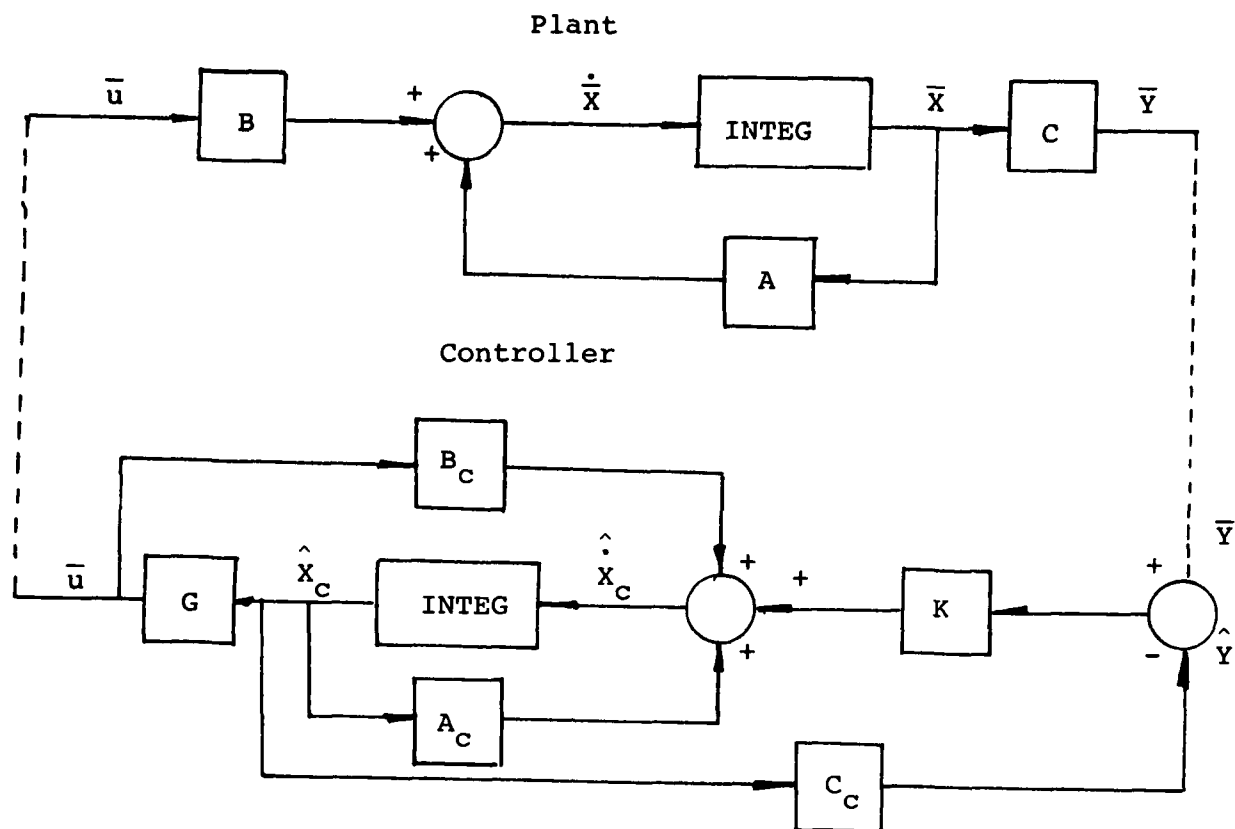


Figure 4. System in Block Diagram Form

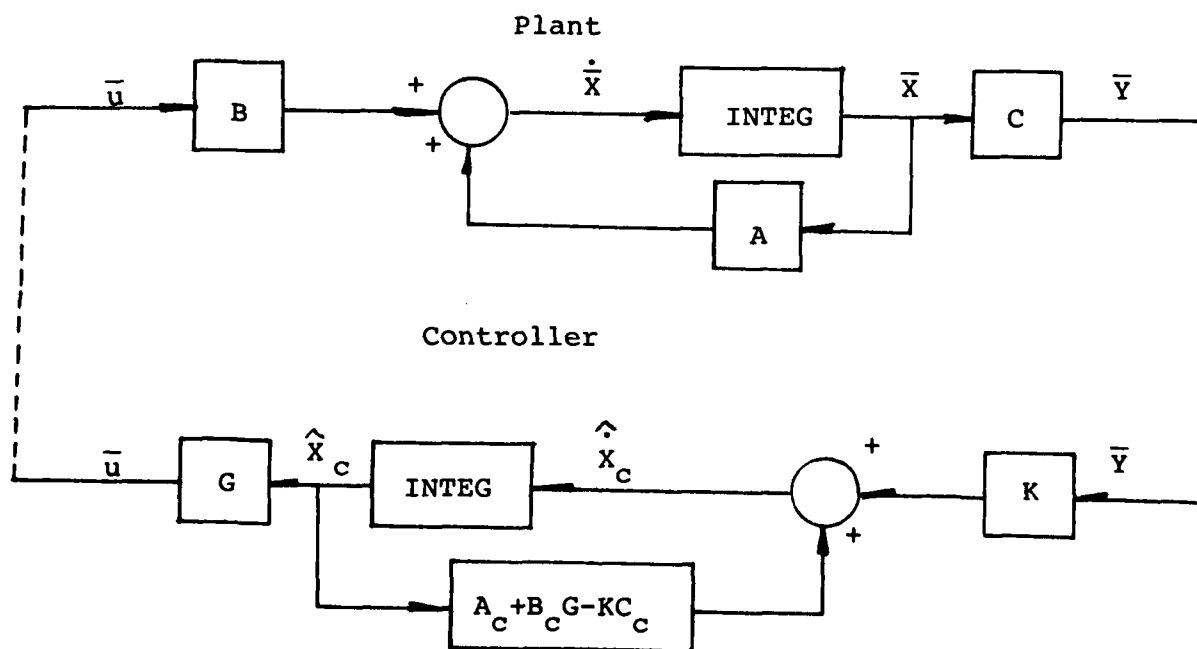


Figure 5. System in Modified Block Diagram Form

produce a stable system, the potentially disastrous effects of an intermittent decoupling must be emphasized. An examination of the eigenvalues of $(A_c + B_c G - KC_c)$ will, therefore, be included in the analysis.

Transformation Matrix Control

It has been shown in what has preceded that, due to observation spillover and control spillover, the system represented by Eq (44) could be made unstable. In an attempt to alleviate this problem we will employ a control technique which attempts to eliminate spillover. This suggests driving the off-diagonal matrices of Eq (44) to zero, while retaining active feedback control of the overall system. An examination of the system equation leads one to realize that, if either $B_s G$ or KC_s are zero, the system eigenvalues revert to the eigenvalues falling on the diagonal. The nature of these diagonal matrices is such that negative eigenvalues (and, hence, system stability) are guaranteed. Obviously, one solution to $B_s G = 0$ is $G = 0$. However, this solution also renders $B_c G = 0$, and control is forgone. That being the case, the transformation method is directed at constraining the feedback gain matrices such that:

$$B_s G = 0 \quad (50)$$

$$KC_s = 0 \quad (51)$$

while, at the same time:

$$B_c G \neq 0 \quad (52)$$

$$KC_c \neq 0 \quad (53)$$

To develop this method, we will first look at the conditions required to satisfy Eqs (50) and (52), namely the elimination of control spillover. At the core of this method will be

an attempt to find some transformation matrix, T , such that subsequent control vector, $\bar{U}(t)$, required for Eq (6) will be:

$$\bar{U}(t) = T\bar{z}(t) \quad (54)$$

where $\bar{z}(t)$ is now the new control input and with constraint that:

$$B_s^T = 0 \quad (55)$$

while: $B_c^T \neq 0 \quad (56)$

One method with which to obtain this transformation matrix employs a technique known as Singular Value Decomposition (Ref 9). Using SVD allows reformulation of the $s \times m$ B_s matrix as:

$$B_s = W\Sigma V^T \quad (57)$$

where W is an $s \times s$ orthogonal matrix of left singular vectors

V is an $m \times m$ orthogonal matrix of right singular vectors

and

$$\Sigma = \begin{bmatrix} S & \vdots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \vdots & 0 \end{bmatrix}_{s \times m} \quad (58)$$

Such that S is a $q \times q$ diagonal matrix of the singular values of B_s , or: (continued)

$$S = \begin{bmatrix} \sigma_1 & 0 & . & . & . & . & . & 0 \\ 0 & \sigma_2 & . & . & . & . & . & 0 \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ 0 & . & . & . & . & . & . & \sigma_q \end{bmatrix}$$

Singular values are always greater than or equal to zero, and the total number of non-zero singular values is equal to the rank of the decomposed matrix. As long as B_s is full rank with dimensions of $s \times m$, then q is the minimum value of the pair (s,m) . By arbitrarily letting r be the difference between q and m , or:

$$q + r = s \quad (59)$$

The W can be partitioned such that:

$$W = \begin{bmatrix} W_q & \vdots & W_r \end{bmatrix}_{s \times s} \quad (60)$$

having:

W_q as an $s \times q$ matrix

W_r as an $s \times r$ matrix

In a similar fashion, we can choose p as the difference between q and n such that:

$$q + p = m \quad (61)$$

We can now partition the right singular vector matrix, V , as:

$$V = \begin{bmatrix} V_q & \vdots & V_p \end{bmatrix}_{m \times m} \quad (62)$$

having:

V_q as an $m \times q$ matrix

V_p as an $m \times p$ matrix

By defining V_p as our transformation matrix, T , we find some highly desirable results with respect to Eqs (55) and (56), namely:

$$B_s^T = B_s V_p = W_q S V_q^T V_p \quad (63)$$

However, since V is an orthogonal matrix:

$$V_q^T V_p = 0 \quad (64)$$

Hence:

$$B_s^T = 0 \quad (65)$$

Coordinating this expression for the transformation matrix with the model so far established, it should first be noted that the dimensions of B_s are directly the result of both the number of modes to be suppressed (s) and the number of actuators employed (m). If the rank of the matrix B_s is equal to the number of actuators available, then $q = m$. By Eq (61) it is seen that this forces p to be zero. It follows that $V_p = T = 0$, and we are restricted to the trivial solution. Recalling the previous commitment for the transformation method, this would fail by allowing $B_c^T = 0$. It is clear that for the transformation method to be carried to an exact solution, special conditions including $q < m$ must be met. Restated, the rank of B_s must be less than the number of actuators. It should, however; be noted that if you are restricted to a fixed number of pre-oriented actuators, performance is enhanced by using the singular vector associated with the least singular values (even

though $B_s T \neq 0$). In any case, where $q = m$, an order reduction scheme is required to get an exact non-trivial solution to $B_s T = 0$. As a verifying set of examples, let q be the rank of B_s , and let that matrix have dimensions $s \times m$ with $s = 4$ and $m = 3$. If B_s is full rank, $q = 3$. Therefore by Eq (61), $p = 0$. However, if we can reduce the rank of B_s to $q = 2$, then $p = 1$ and V_p is non-zero. As will be demonstrated, this rank deficiency is obtained either through judicious orientation of the actuators (driving a non-zero singular value to zero) or through addition of actuators and increasing m . The minimum number of actuators that can be used where the former method is employed is two, since a matrix of rank 1 cannot be made rank deficient.

Regardless of how an appropriate non-zero T is formed, we will now have the resultant solution vector in Eq (54), where:

$$\bar{z} = -\bar{G}_t \bar{X}_c \quad (66)$$

and

$$R_t = T^T R T \quad (67)$$

such that R_t is a $p \times p$ positive definite matrix, and:

$$B_t = B_c T \quad (68)$$

with $A_c B_t$ completely controllable.

To generate the new control vector, \bar{z} , the same approach as followed in Section III is employed. The control gain matrix is now defined by:

$$G_t = R_t^{-1} B_t^T P_c \quad (69)$$

and

$$P_C A_C + A_C^T P_C + P_C B_t R_t^{-1} B_t^T P_C + Q_C = 0 \quad (70)$$

where:

G_t is a $p \times m$ reduced degree of freedom critical state
feedback gain matrix

P_C is a $m \times m$ positive definite solution to the reduced
order Ricatti Equation

The gain matrix is therefore finally transformed by:

$$G_C = T G_t \quad (71)$$

which will produce a new m dimensional control with zero control
spillover.

A parallel technique is employed to eliminate the KC_s
observation spillover term. Here, the number of sensors must
exceed the number of suppressed modes, or C_s must be made rank
deficient through sensor re-orientation. The specific methodo-
logy for reducing the order of the optimal regulator will be
described as part of the computer model, and in the investigation
which follows.

Computer Model

Appendix A represents a computer listing for one run of the main program. This particular run applies the transformation method to an eight mode nominal model such that the control spill-over ($B_s G$) is driven to zero. Although the program is seen to be quite lengthy, the comment cards which have been included for clarity suggest the overall straightforwardness of the approach.

As a first step, the parameter matrices (A, B, C) are built. The A matrix portion of the program reads in the natural frequencies from the NASTRAN data, and uses these frequencies and a prescribed damping ratio (0.005) to fill this parameter matrix appropriately. The B and C matrices are formed as a matrix product of mode shapes and actuator or sensor locations. That is, B is formulated as:

$$B = \phi^T D \quad (72)$$

where

ϕ is the matrix whose columns are the eigenvectors
for each mode supplied by NASTRAN (mode shapes)

D is a direction cosine matrix for the locations
and orientations of the prescribed actuators.

Since the sensors and actuators are collocated, it then becomes clear that:

$$C = B^T = D^T \phi \quad (73)$$

Next, by supplying as an input value the number of modes to be controlled, the program takes the A, B, and C matrices and

generates their controlled and suppressed counterparts (i.e. A_c , A_s, B_c ...). With these matrices formed, along with their transposes, the steady state feedback gain matrices (K and G) are established. Reviewing this process as described in Section III, it is seen that one step involves solution of the steady state matrix Ricatti Equation. This solution is obtained via highly specialized computer subroutines created by Kleinman (Ref 9). With these gain matrices, the total system equation seen as Eq (44) is formed. An eigenvalue analysis using subroutine EIGRF from the International Mathematical and Statistical Library (IMSL) is completed against the controller ($A + B_c G$) the observer ($A - K C_c$), and the entire system. This allows for a stability analysis based on these eigenvalues.

Next, a time history response (20 seconds) is performed on the line of sight in both the x and the y directions at grid point 1. This is accomplished in two steps. First, the CC6600 subprogram library of the Air Force Institute of Technology is implemented such that program ODE (Ref 10) can be used to integrate the state equation:

$$\dot{\bar{X}} = A\bar{X} + B\bar{u} \quad (74)$$

to establish $x(t)$ for $t = 0.0, 0.1, 0.2, \dots, 20.0$. Then, using the mode shapes and, primarily their x and y components at grid point 1, the line of sight magnitudes are formulated such that:

$$X_1(t) = \sum_{i=1}^n \phi_{1i} x_i(t) \quad (75)$$

and

$$x_2(t) = \sum_{i=1}^n \phi_{2i} x_i(t) \quad (76)$$

with

$x_1(t)$ being the line of sight in the x direction

$x_2(t)$ being the line of sight in the y direction

This set of results provides a baseline for comparison of future analyses. Once these plots are completed, a singular value decomposition is performed on B_s as the first step in the transformation method. The actuator corresponding to grid point 7 is rotated incrementally until the least singular value of B_s becomes nearly zero. In effect, this reduces the rank of B_s . With this new orientation, a new control gain matrix (G) is formed using the methods described in Section III. Also, new B and C matrices are created to account for the re-oriented sensor/actuator pair.

With these new values, the program returns to the eigenvalues obtained previously. The fact that the system eigenvalues are those of the diagonal members is born out. New plots are then generated so as to compare the time history responses with and without control spillover. This same approach is followed in driving the observation spillover (KC_s) to zero. This set of runs demonstrates the improvement available without adding hardware.

Finally, two sensors are added at grid point one to examine the effectiveness of adding some fairly simple hardware (as opposed to adding actuators). These two sensors are given an

orientation typical to those sensors already prescribed. This run is repeated against a twelve mode model to verify the legitimacy of the first truncation of higher modes. It should be noted that the selection of two additional sensors at grid point one was arbitrary. Any number of additional sensors could be added at any location for this final study.

Investigation

Outline

A systematic approach toward assessing the effectiveness of the transformation method was initiated. As a first case, the system eigenvalues and line of sight time history responses were examined for models with and without control spillover ($B_s G$). The transformation technique was only applied to the control gain (G). The sensor/actuator pairs remained collocated, while one of these pairs was rotated to produce an additional zero singular value to B_s . An angular orientation was obtained which produced a rank reduction in the suppressed control matrix. The weighting function of the controlled states, \bar{X}_c , was set at the identity matrix. Upon successful completion of this first case, the process was repeated with increasingly higher control weighting. Then, this set of runs was compared to the case of eliminating observation spillover ($K C_s$), rather than control spillover. The purpose of this alteration to the main program is twofold. First, it would demonstrate that the total system matrix (Eq (44)) is block diagonalized successfully by forcing either of the spillover terms to zero. Second, it facilitates the final area of investigation; namely the potential benefits of sensor additions. The addition of sensors (rather than actuators) within the prescribed model was chosen out of practicality. From a "hardware" viewpoint, the addition of sensors is seen to be considerably more realistic than the addition of point force actuators.

For all cases examined, the overall attempt is to reduce the line of sight error in the x and y directions at grid point one to less than 0.0004 radians and less than 0.00025 radians respectively in 20 seconds.

Elimination of Control Spillover

In an attempt to further clarify the direction of this analysis recall from Sections II and III that the control gain matrices are determined using steady state optimal regulator theory, which involves minimization of related quadratic performance indices. These performance indices for the model with and without spillover are:

$$J = \frac{1}{2} \int_0^{\infty} (\bar{X}_c^T F \bar{X}_c + \bar{u}^T R u) dt$$

$$J = \frac{1}{2} \int_0^{\infty} (\bar{X}_c^T F \bar{X}_c + \bar{U}_t^T R_t \bar{U}_t) dt$$

respectively.

An inspection of these two indices demonstrates the role of the control weighting matrix, F , as an amplifier of the resultant gains applied to the controlled states. All cases run attempt to control the first two modes and suppress the remaining six. An attempt to modify, and ultimately improve performance is tied to increasing the magnitude of this weighting matrix. It is known from the previously developed theory that increasing the magnitude of the control gain (here G) has the coincident negative effect of increasing control spillover. It is with this awareness that the first study is accomplished. This study involves generating the system matrix and examining

the eigenvalues and associated line of sight time history response. Once these data are generated, the transformation technique of Section III is applied to force $B_s G$ to zero.

Table 4 is a presentation of data pertinent to the first case in which F is set at the identity matrix. Both sets of system eigenvalues exhibit stability. Additionally, eigenvalues of the entire system are the same as the eigenvalues of the matrices on the diagonal, verifying that control spillover has been eliminated. It should also be noted that the transformation method has generated a controller $(A + B_c G - KC)$ that is unstable, but which, none-the-less, produces a stable system. Figures 6 through 9 represent the time history responses for the x line of sight and y line of sight errors. Although a precise bandwidth on the error is difficult to establish, it is obvious that the prescribed limits specified in the outline portion of this section have not been satisfied.

The next step then involves multiplying the F matrix by scalar powers of ten (i.e. 1, 10, 100,...). Until F reaches $1000[I]$, there is no significant improvement in the line of sight error for either the case with or without control spillover. However, at the F 1000 $[I]$, significant changes in the system response become evident. Table 5 is a presentation of the associated eigenvalues for this case. Clearly, the spillover terms have now forced the system (without transformation of the control gain) unstable. The eigenvalues after gain transformation, however, still exhibit stability. This demonstrates the certain advantages of using this method. Figures 10 and 11

depict line of sight errors without transformation and demonstrates the unstable response indicated by the associated eigenvalues. The time histories of Figures 12 and 13 present an x line of sight error for the system with $B_s G = 0$ within an approximate bandwidth of ± 0.0013 , and a y line of sight error of ± 0.0008 . As the control gain weighting function is increasing there is no significant improvement of response. The trend of these data suggests that the criteria for pointing accuracy cannot be met with the prescribed number of sensors and actuators (6 each). It is clear, for the reasons highlighted, that sensors will have to be added.

Table IV

Elimination of Control Spillover; $F = 1.0[I]$

<u>System Eigenvalues</u>		
<u>Before</u> <u>Transformation ($B_s G \neq 0$)</u>		<u>After</u> <u>Transformation ($B_s G = 0$)</u>
$-.02822 \pm 5.70935i$	-S	$-.02855 \pm 5.71073i$
$-.02838 \pm 5.67583i$	-S	$-.02838 \pm 5.67583i$
$-.02553 \pm 5.14848i$	-S	$-.02575 \pm 5.14935i$
$-.01918 \pm 3.84804i$	-S	$-.01924 \pm 3.84834i$
$-.01778 \pm 3.55770i$	-S	$-.01778 \pm 3.55770i$
$-.01467 \pm 2.96372i$	-S	$-.01482 \pm 2.96458i$
$-.08663 \pm 1.47902i$	-C	$-.07712 \pm 1.46602i$
$-.06679 \pm 1.18915i$	-C	$-.00751 \pm 1.17064i$
$-.06279 \pm 1.45703i$	-O	$-.08627 \pm 1.46583i$
$-.03768 \pm 1.16069i$	-O	$-.04420 \pm 1.17052i$

<u>Eigenvalues of $A_c + B_c G$</u>		
$-.07457 \pm 1.46607i$	-C-	$-.07712 \pm 1.46602i$
$-.05199 \pm 1.17046i$	-C-	$-.00751 \pm 1.17064i$

<u>Eigenvalues of $A_c - KC_c$</u>		
$-.07457 \pm 1.46607i$	-O-	$-.08627 \pm 1.46583i$
$-.05199 \pm 1.17046i$	-O-	$-.04420 \pm 1.17052i$

<u>Eigenvalues of $A_c + B_c G - KC_c$</u>		
$-.00733 \pm 1.46222i$		$.00194 \pm 1.46119i$
$-.00585 \pm 1.16818i$		$.03072 \pm 1.17082i$

C = Controlled Mode Eigenvalues

S = Suppressed Mode Eigenvalues

O = Observer Mode Eigenvalues

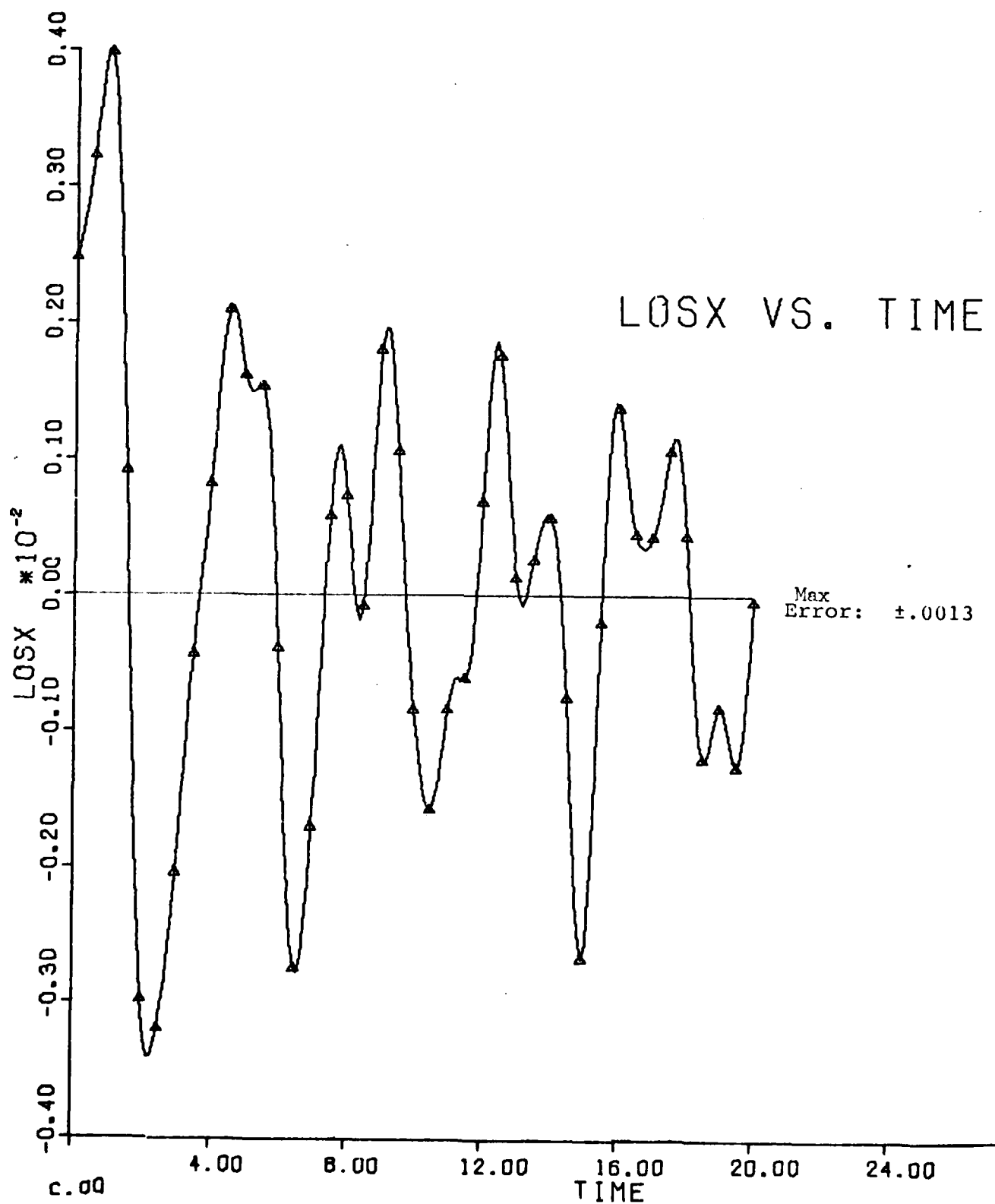


Figure 6. LOSX VS. TIME, $B_S G \neq 0$, $F = 1.0$

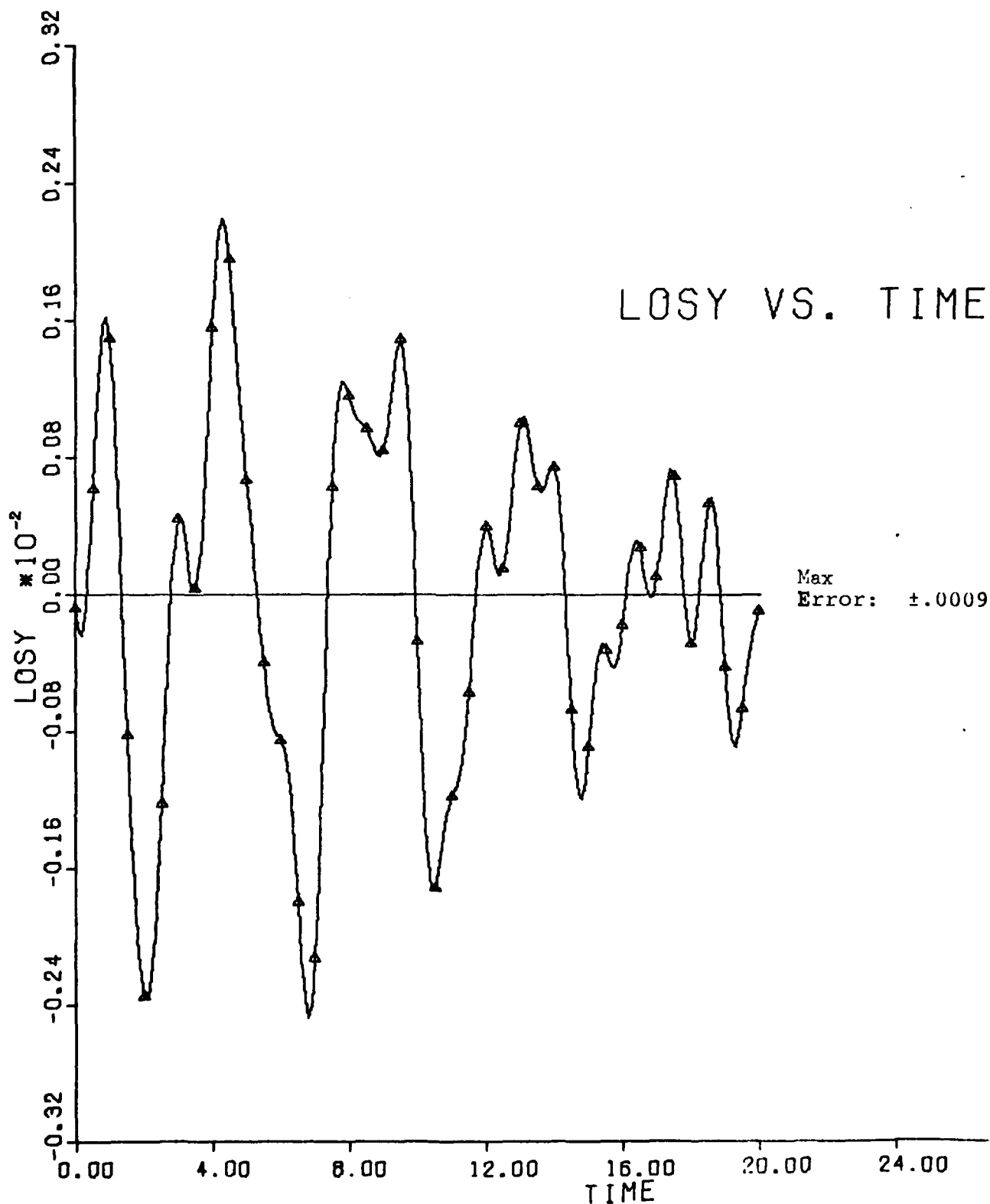


Figure 7. LOS Y VS. TIME, $B_S \neq 0$, $F = 1.0$

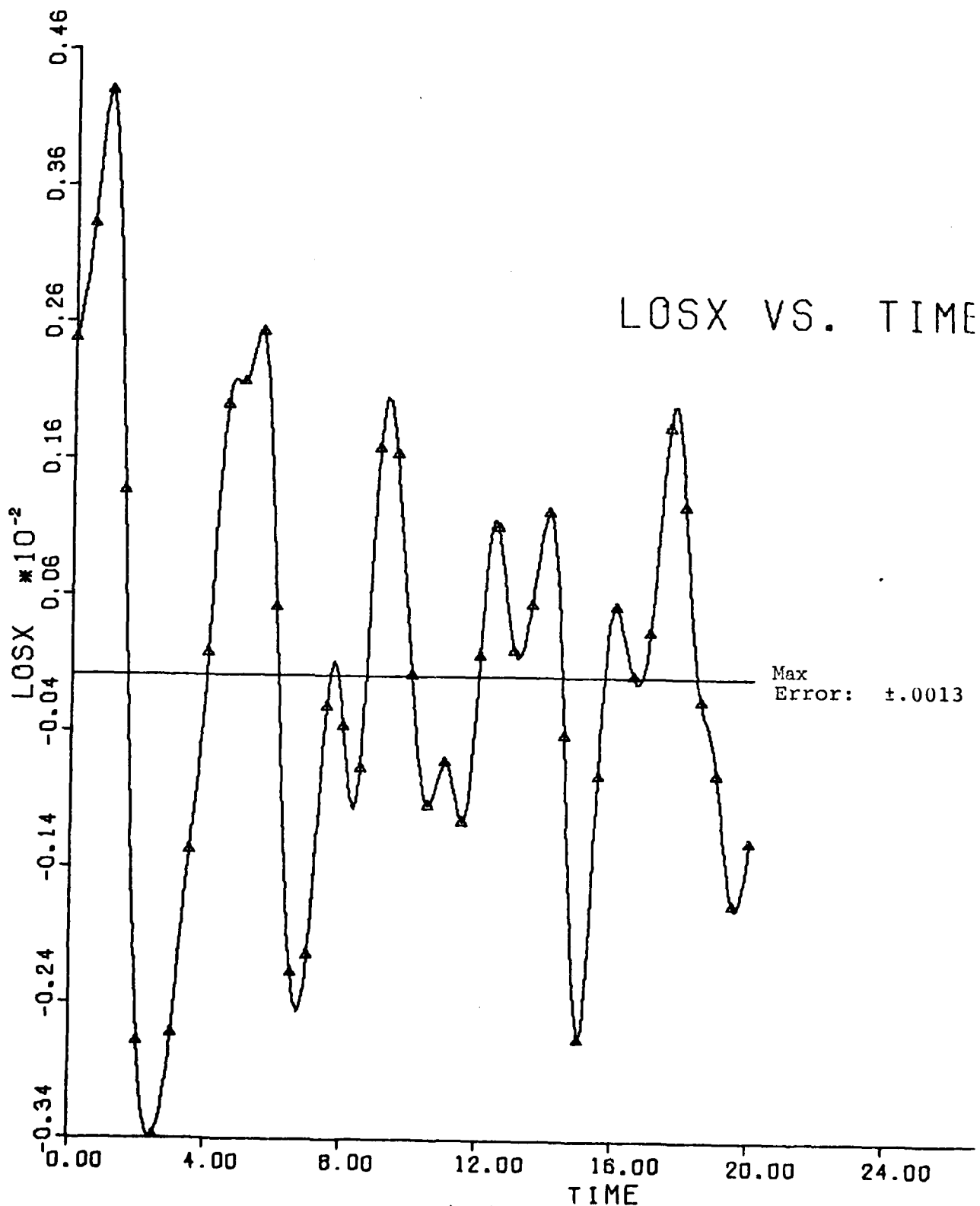


Figure 8. LOSX VS. TIME, $B_S = 0$, $F = 1.0$

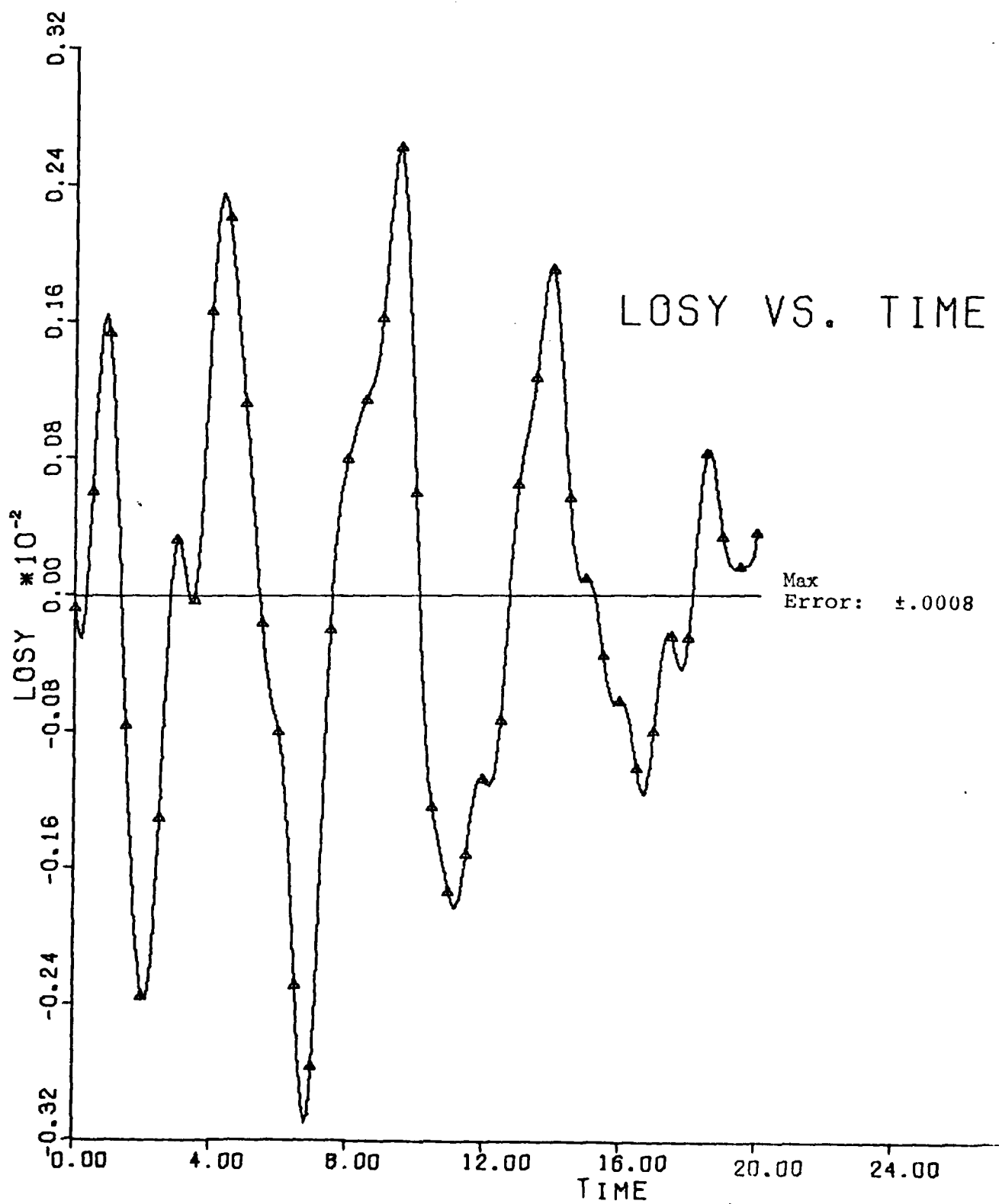


Figure 9. LOSY VS. TIME, $B_s G = 0$, $F = 1.0$

Table V

Elimination of Control Spillover; $F = 1000.0[I]$ System Eigenvalues

<u>Before Transformation ($B_S G \neq 0$)</u>		<u>After Transformation ($B_S G = 0$)</u>	
.09178 ± 5.73850i	-S-	-.02855 ± 5.71073i	
-.02838 ± 5.67583i	-S-	-.02838 ± 5.67583i	
.03098 ± 5.17892i	-S-	-.02575 ± 5.14935i	
-.00835 ± 3.85687i	-S-	-.01924 ± 3.84834i	
-.01764 ± 3.55783i	-S-	-.01778 ± 3.55770i	
.01328 ± 2.98115i	-S-	-.01482 ± 2.96458i	
-.300048 + 0i	-C-	-3.11793 + 0i	
-1.15711 + 0i	-C-	-1.46147 + 0i	
-1.66482 ± .80887i	-C-	-.02000 ± 1.17188i	
-.07067 ± 1.46398i	-O-	-.08627 ± 1.46583i	
-.04480 ± 1.16631i	-O-	-.04420 ± 1.17064i	

Eigenvalues of $A_C + B_C G$

-2.89759 + 0i	-C-	-3.11793 + 0i
-1.53062 + 0i	-C-	-1.46147 + 0i
-1.5091- ± .74849i	-C-	-.02000 ± 1.17188i

Eigenvalues of $A_C - K C_C$

-.07458 ± 1.46608i	-O-	-.08627 ± 1.46583i
-.05199 ± 1.17046i	-O-	-.04420 ± 1.17064i

Eigenvalues of $A_C + B_C G - K C_C$

-3.02017 + 0i		-3.23401 + 0i
-1.46304 ± .64937i		-1.18781 + 0i
-1.27355 + 0i		-.01849 ± 1.17817i

C = Controlled Mode Eigenvalues

S = Suppressed Mode Eigenvalues

O = Observer Mode Eigenvalues

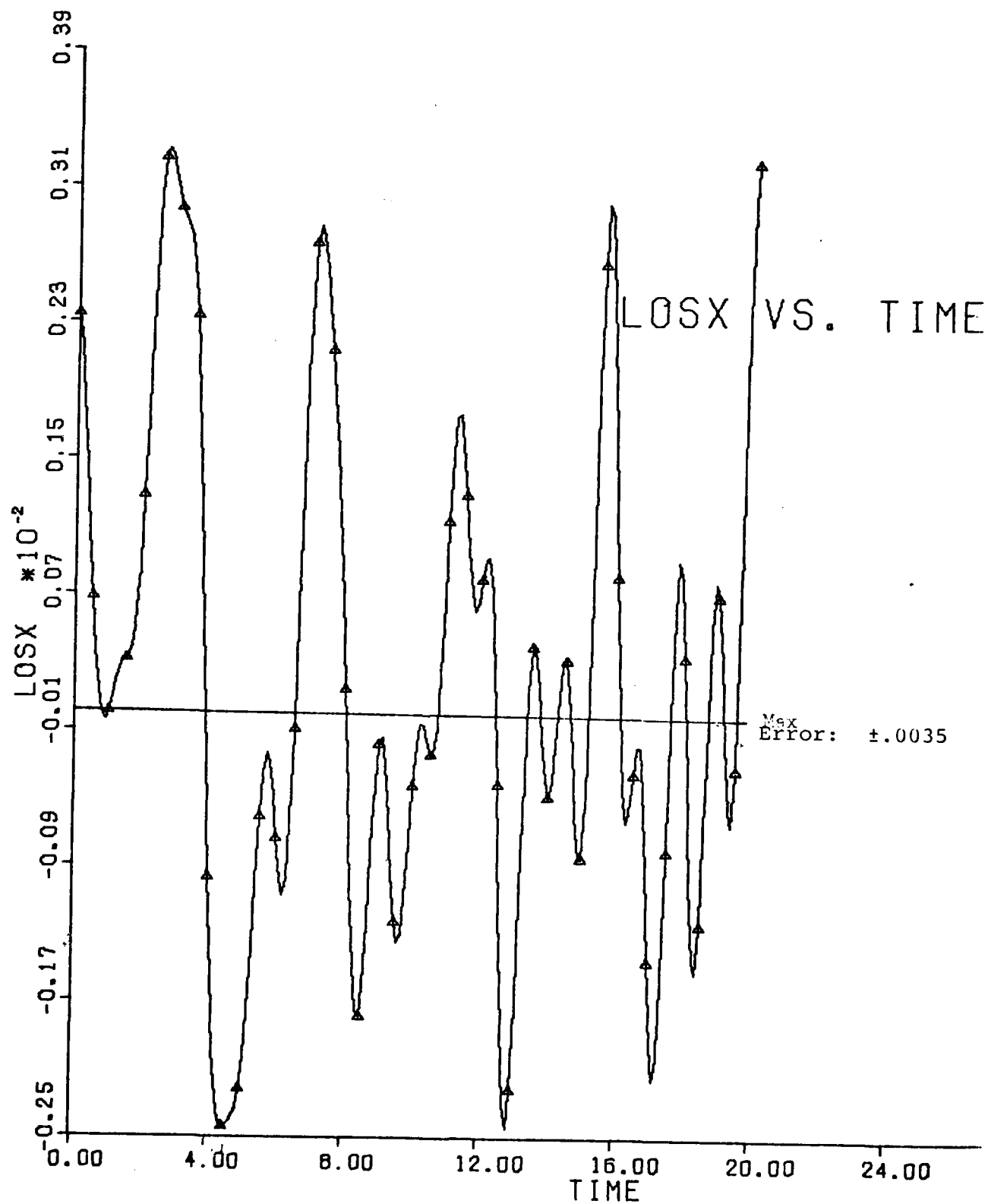


Figure 10. LOSX VS. TIME, B_S G ≠ 0. F = 1000.0

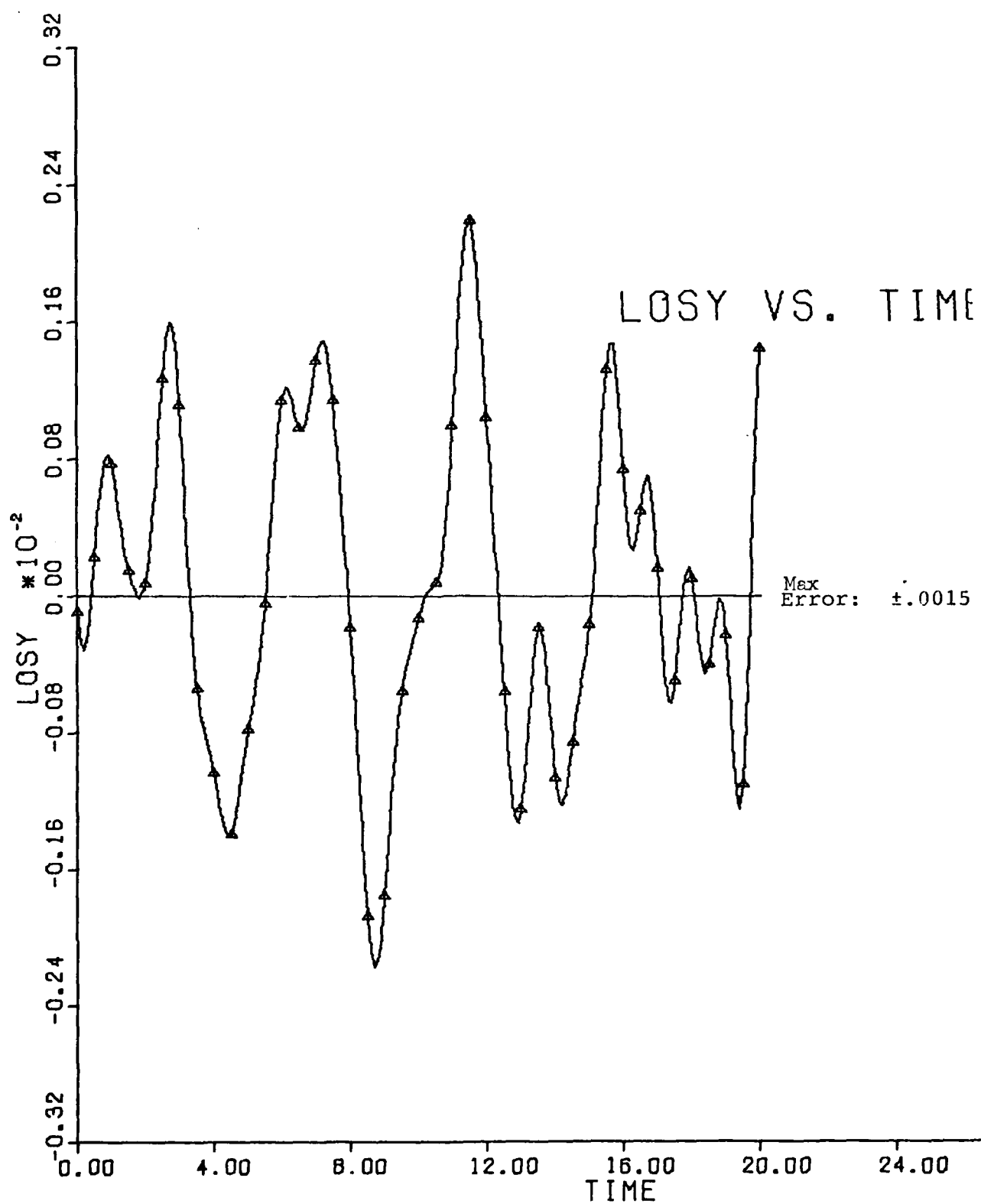


Figure 11. LOS Y VS. TIME, $B_S \neq 0$, $F = 1000.0$

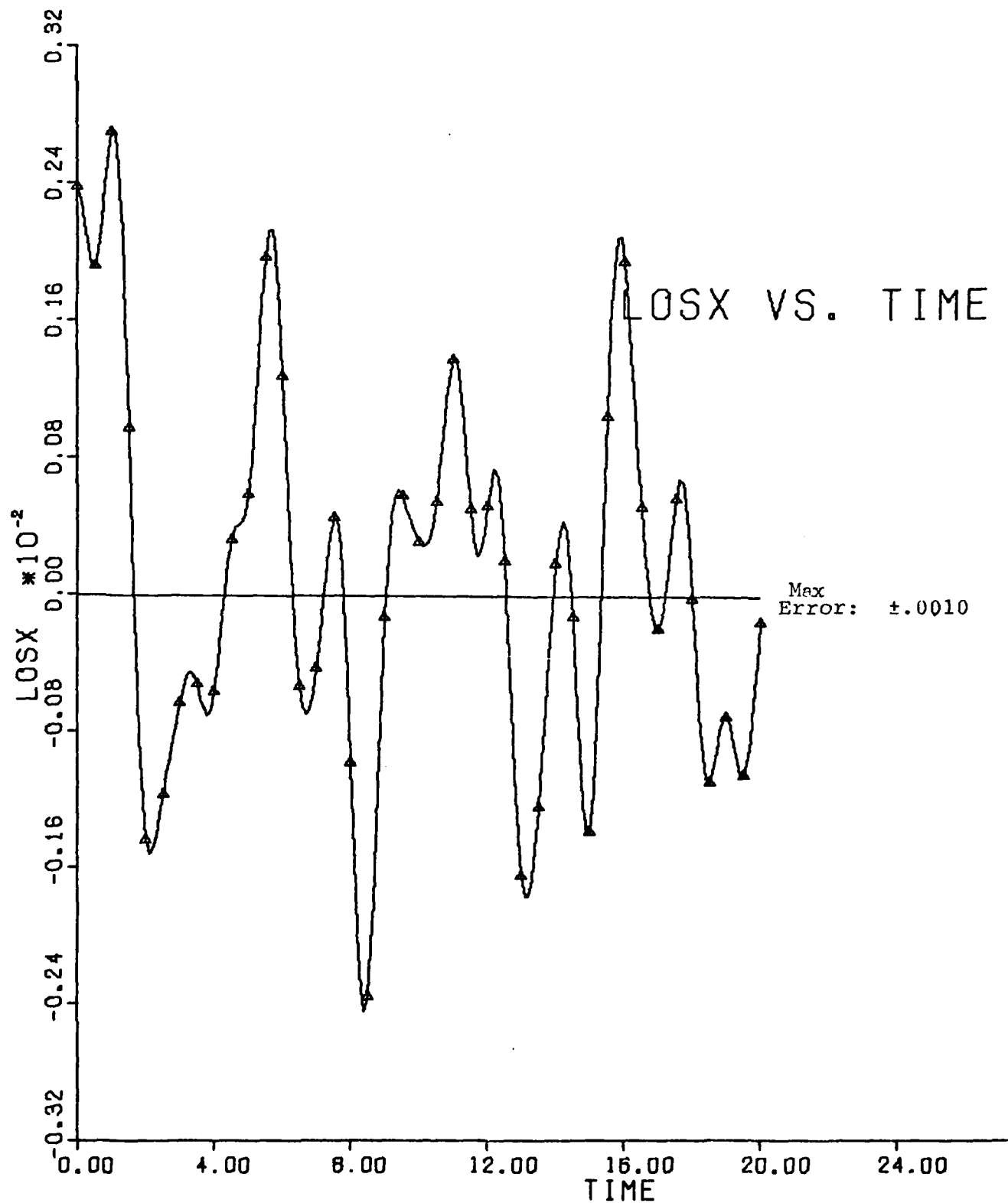


Figure 12. LOSX VS. TIME, $B_s G = 0$, $F = 1000.0$

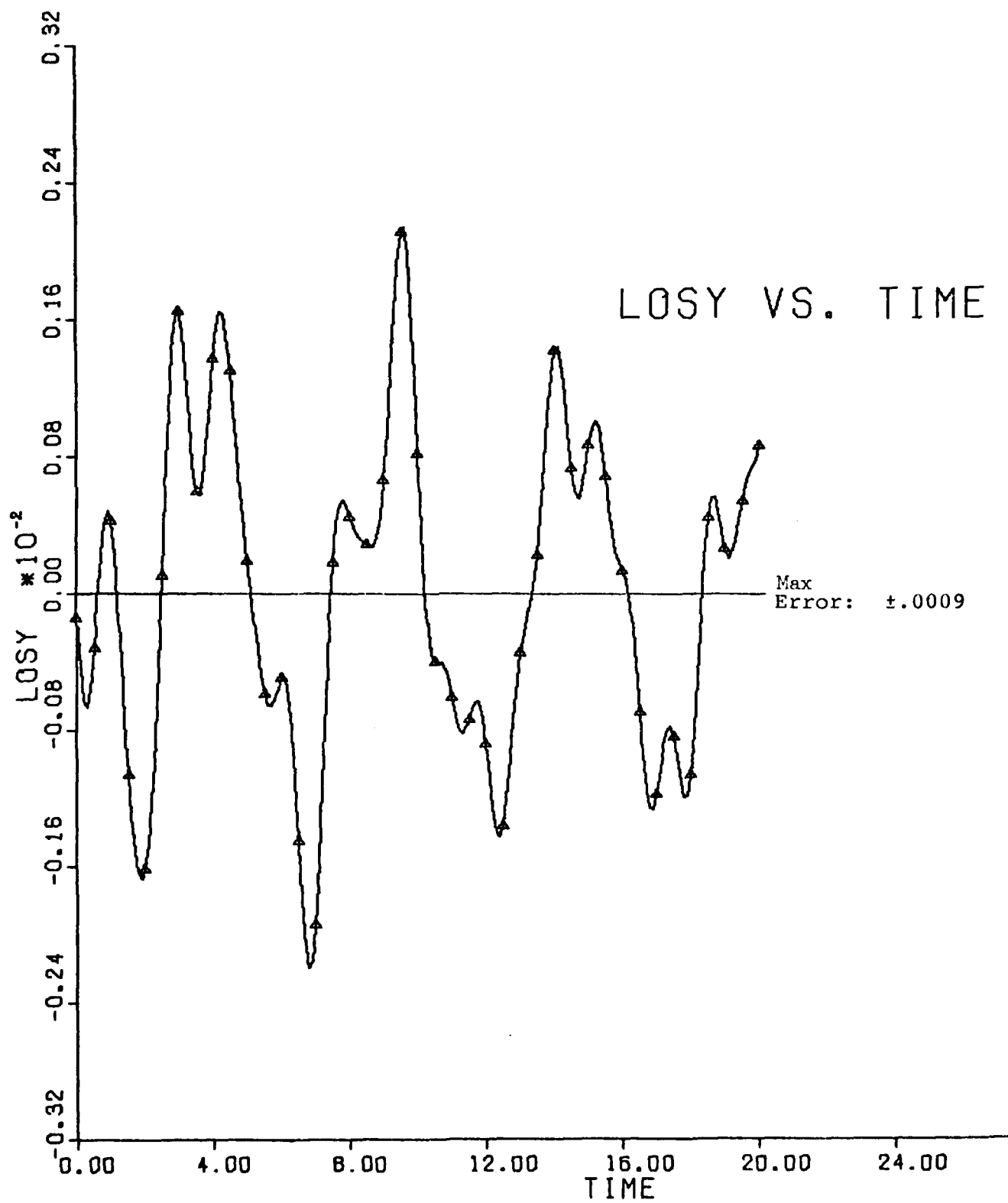


Figure 13. LOS Y VS. TIME, $B_s G = 0$, $F = 1000.0$

Elimination of Observation Spillover

The approach taken during this portion of the analysis is directed by an awareness, a priori, that sensors will be added. As a preliminary step, the procedure for applying the transformation method to the control gains is first reapplied to the observation gain (K). The same sensor and actuator pair at grid point seven is again rotated until the smallest singular value of C_s is driven to zero. Table VI presents the results of the eigenvalue analysis which followed. Q replaces F as the observation weighting matrix acting on the controlled states. Once again, the system matrix is seen to be stabilized and diagonalized via the transformation method. A more pertinent case, in light of a forthcoming examination of sensor additions, is an application of the transformation method to the system with the sensors in their original orientation. Table VII, below, lists the singular values of C_s for the fixed six sensors. An examination of their relative magnitudes indicates that, although the last singular value is non-zero, some potential benefits may be gained by applying the transformation method with this singular value and its associated right singular vector.

Table VII

Singular Values of C_s ; Six Non-reoriented Sensors

<u>Number</u>	<u>Singular Value</u>
1.	.70706
2.	.70423
3.	.70363
4.	.49803
5.	.42875
6.	.28536

Table VI

Elimination of Observation Spillover; $Q = 1000.0 \begin{bmatrix} I \end{bmatrix}$ System Eigenvalues

<u>Before Transformation ($KC_s \neq 0$)</u>		<u>After Transformation ($KC_s = 0$)</u>	
.12469 ± 5.69223i	-S-	-.02855 ± 5.71073i	
-.02838 ± 5.67583i	-S-	-.02838 ± 5.67583i	
.06669 ± 5.15788i	-S-	-.02575 ± 5.14935i	
.00396 ± 3.85266i	-S-	-.01924 ± 3.84834i	
-.01746 ± 3.55780i	-S-	-.01778 ± 3.55770i	
.03719 ± 2.97542i	-S-	-.01482 ± 2.96458i	
-.06713 ± 1.45779i	-C-	-.08626 ± 1.46583i	
-.03931 ± 1.16093i	-C-	-.04420 ± 1.17052i	
-2.33756 ± .32019i	-O-	-.52896 ± 1.42215i	
-1.72715 ± 1.20427i	-O-	-.05307 ± 1.17791i	

Eigenvalues of $A_c + B_c G$

-.07457 ± 1.46607i	-C-	-.08626 ± 1.46583i
-.05199 ± 1.17046i	-C-	-.04420 ± 1.17052i

Eigenvalues of $A_c - KC_c$

-2.89750 + 0i	-O-	-.52896 ± 1.42215i
-1.50921 ± .74854i	-O-	-.05307 ± 1.17791i
-1.15308 + 0i	-O-	

Eigenvalues of $A_c + B_c G - KC_c$

1.45136 ± .959056i	.40005 ± 1.16987i
2.13224 ± 1.31631i	.43833 ± 1.59171i

C = Controlled Mode Eigenvalues

S = Suppressed Mode Eigenvalues

O = Observer Mode Eigenvalues

Table VIII

Reduction of Observation Spillover; $Q = 1000.0[I]$ (Fixed Sensors)System Eigenvalues

<u>Before</u> <u>Transformation ($KC_s \neq 0$)</u>		<u>After</u> <u>Transformation ($KC_s \neq 0$)</u>
.12469 \pm 5.69223i	-S-	-.06666 \pm 5.71292i
-.02838 \pm 5.67583i	-S-	-.02838 \pm 5.67583i
.06669 \pm 5.15788i	-S-	.09387 \pm 5.21012i
.00396 \pm 3.85266i	-S-	-.05836 \pm 3.81382i
-.01746 \pm 3.55780i	-S-	-.01331 \pm 3.56096i
.03719 \pm 2.97542i	-S-	-.02278 \pm 2.96146i
-.06713 \pm 1.45779i	-C-	-.02024 \pm 1.47982i
-.03931 \pm 1.16093i	-C-	-.09759 \pm 1.10147i
-2.33756 \pm .32091i	-O-	-.27202 \pm 1.34862i
-1.72715 \pm 1.20427i	-O-	-.03520 \pm 1.20825i

Eigenvalues of $A_C + B_C G$

-.07457 \pm 1.46607i	-C-	-.07457 \pm 1.46607i
-.05199 \pm 1.17046i	-C-	-.05199 \pm 1.17046i

Eigenvalues of $A_C - KC_C$

-2.89750 + 0i	-O-	-.20967 \pm 1.17020i
-1.5308 + 0i	-O-	-.04992 \pm 1.46369i
-1.50921 \pm .74854i	-O-	

Eigenvalues of $A_C + B_C G - KC_C$

1.45136 \pm .95906i		-.01187 \pm 1.48607i
2.13224 \pm 1.3163i		.13171 \pm 1.18510i

C = Controlled Mode Eigenvalues

S = Suppressed Mode Eigenvalues

O = Observer Mode Eigenvalues

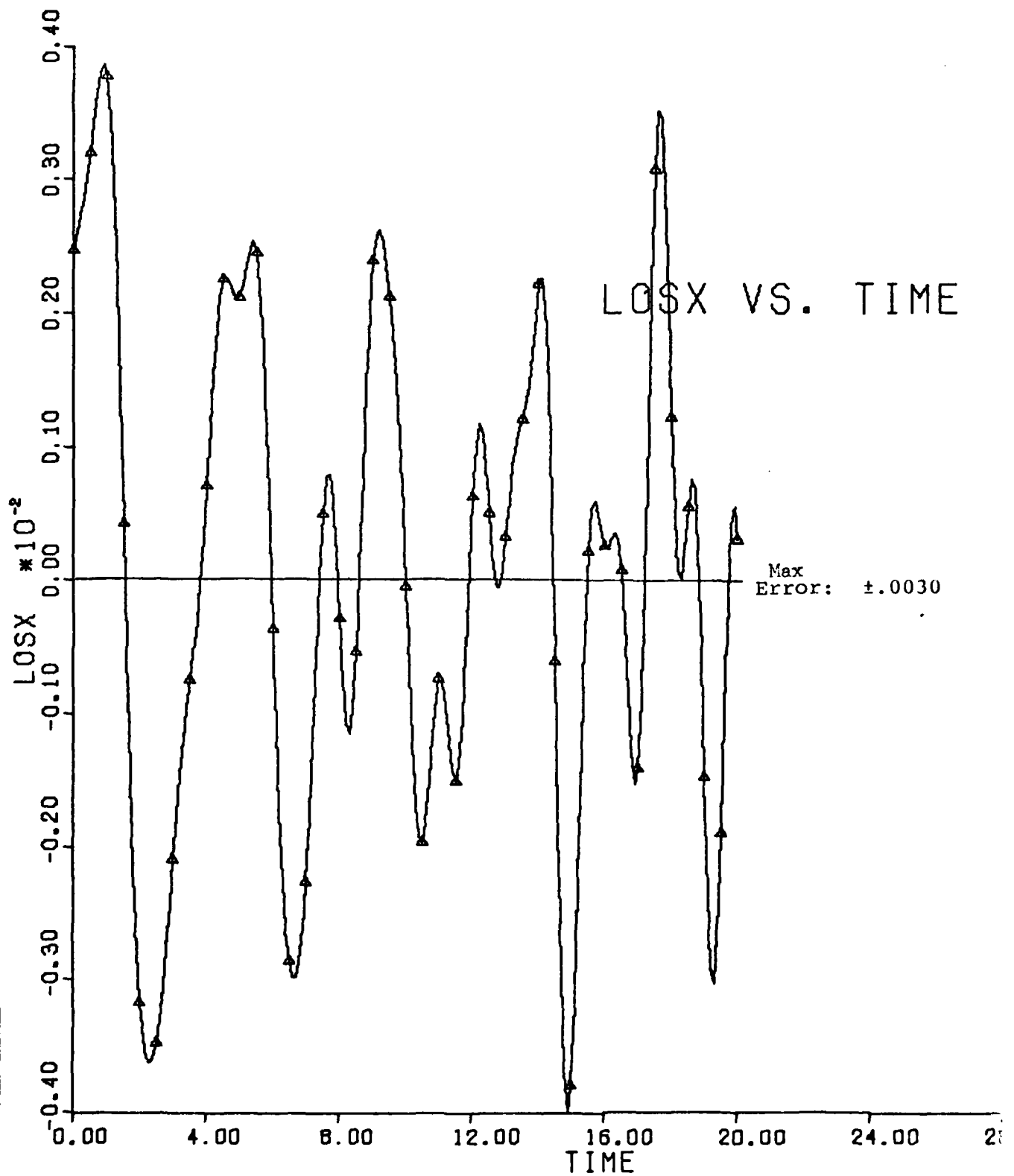


Figure 14. LOSX VS. TIME, $KC_s \neq 0$, $Q = 1000.0$

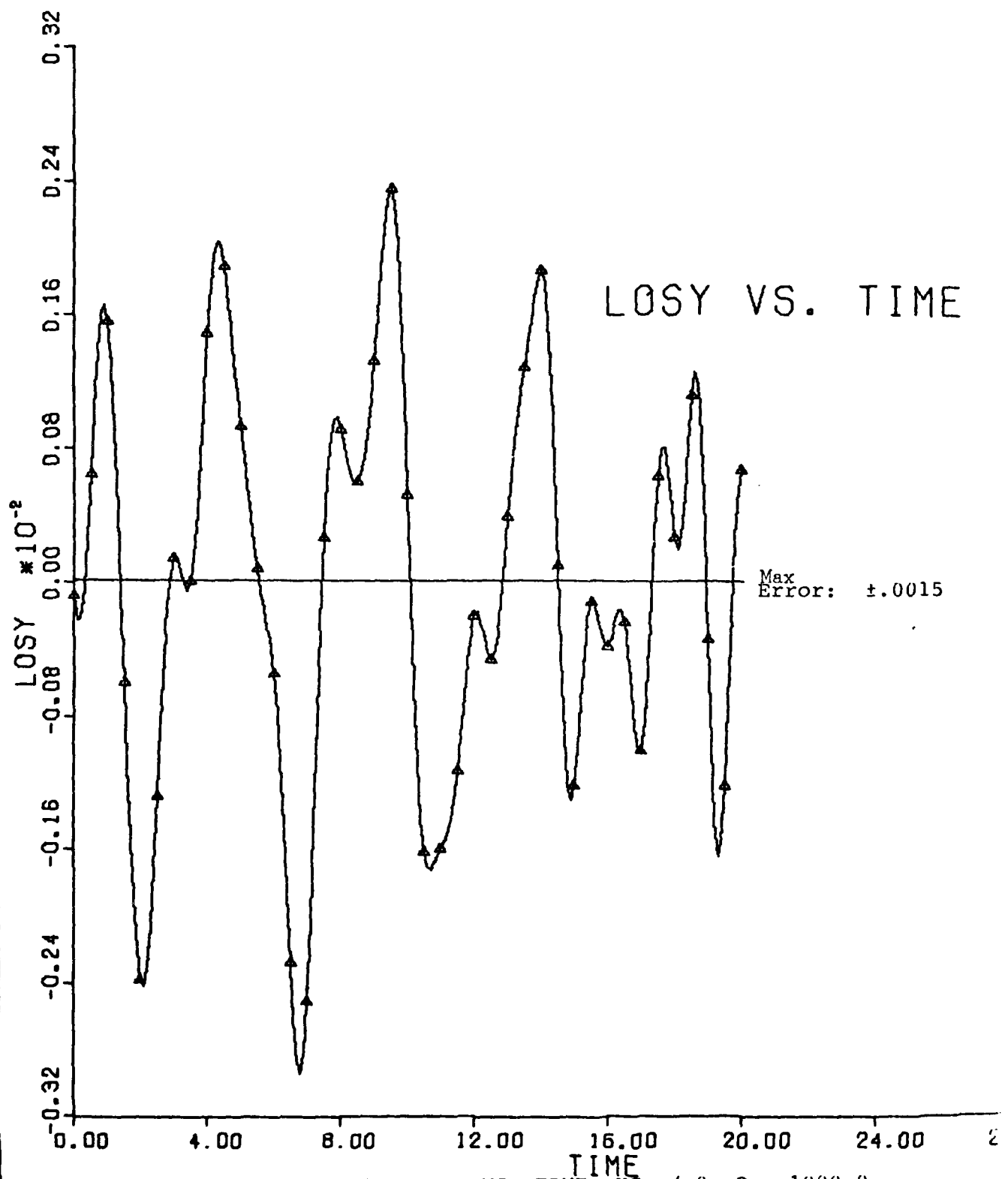


Figure 15. LOS Y VS. TIME, $KC_s \neq 0$, $Q = 1000.0$

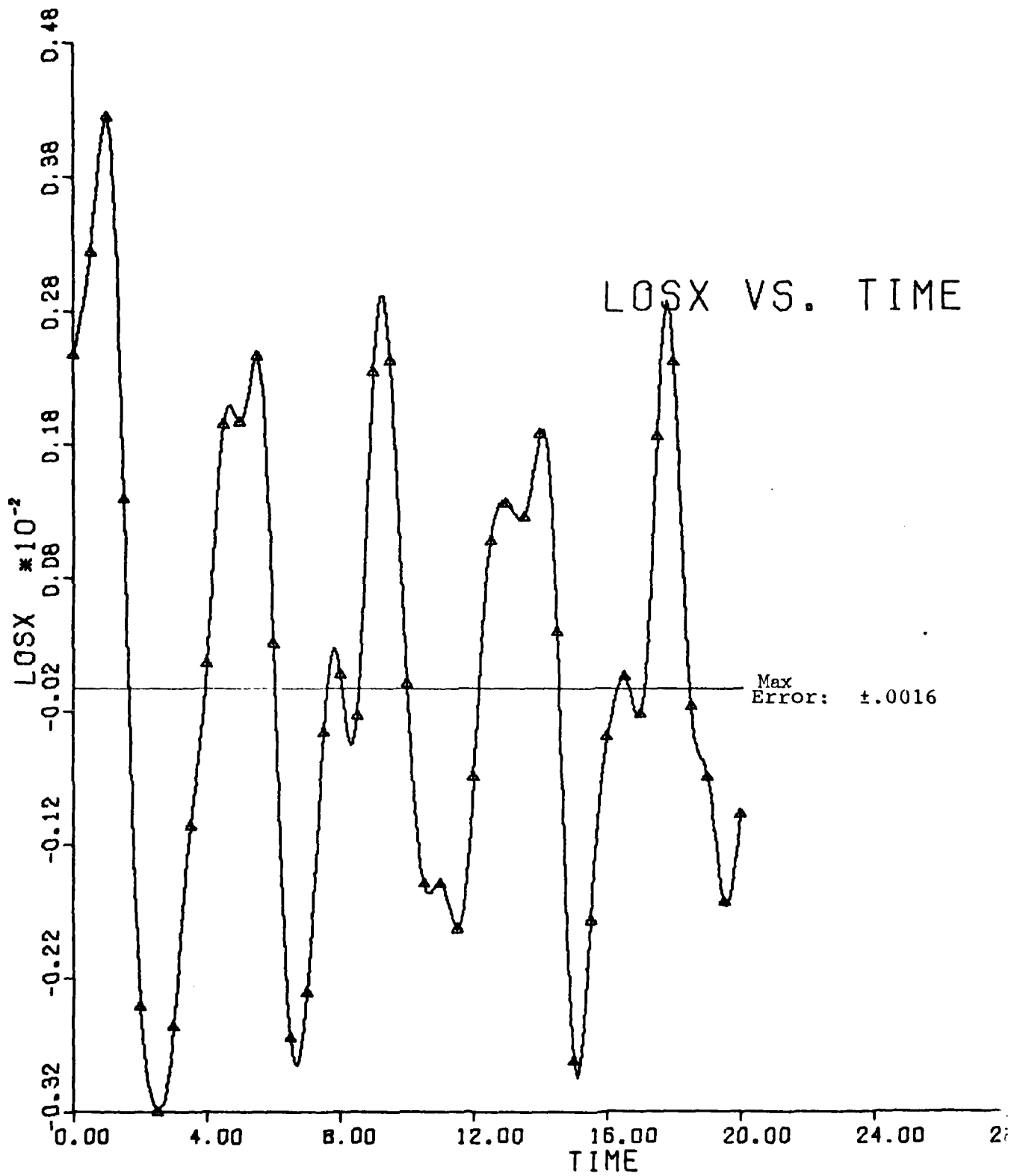


Figure 16. LOSX VS. TIME, $KC_s \neq 0$, $Q = 1000.0$

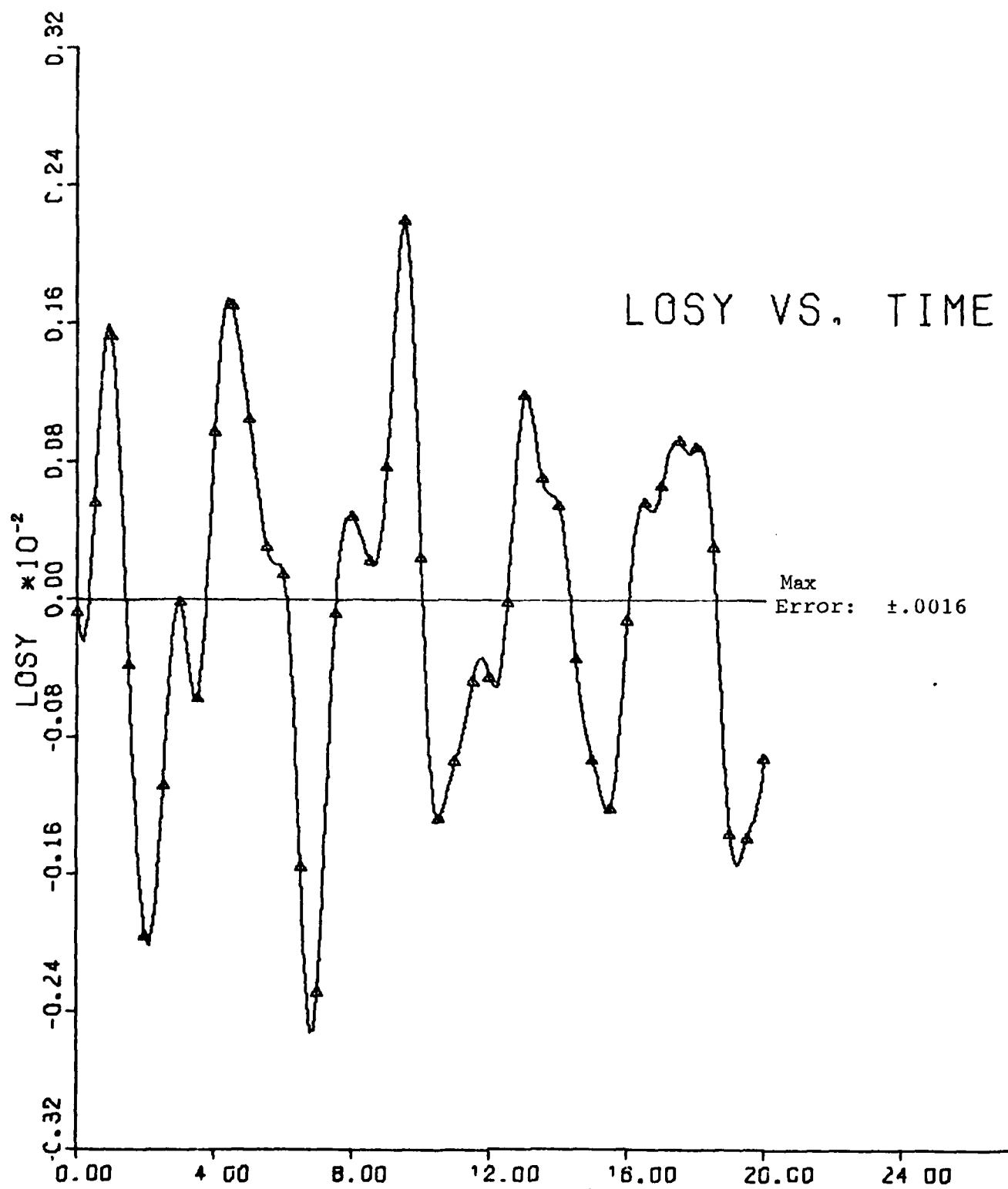


Figure 17. LOS Y VS. TIME, $KC_s \neq 0$, $Q = 1000.0$

The results of this analysis bear out our expectations. First, some observability has been gained, and response is improved. This is born out by examining Figs 14 through 17, which are the line of sight response histories with and without transformation. However, an examination of Table VIII shows that the additional observability was not sufficient to generate a completely stable system. The observation spillover has been reduced, but not eliminated.

Sensor Additions

Until now, we have seen that the transformation method can be successfully applied to the tetrahedron. Spillover can be minimized or completely eliminated, depending on whether or not sensor and actuator reorientations are permitted. In reality, it is perhaps more likely that one would have less than complete liberty to do this. Regardless, the specified line of sight criteria has not been met. Hence, we are left with sensor additions as a last resort.

Two sensors were added to the system at grid point one. This number and location are essentially arbitrary, but will serve as a starting point for more exhaustive subsequent analyses. Table IX, using the format applied throughout this report, presents the results of this case. It is clear that sensor additions have allowed the same system matrix block diagonalization as previous techniques. However, as has been the case previously, Figs 18 through 21 demonstrate that the criteria for line of sight response has not been met. The improvement to note is

between time history response associated with Table VIII and that of Table IX. Clearly, the addition of sensors has enhanced the overall performance.

Table IX

Elimination of Observation Spillover; $Q = 1000.0 \begin{bmatrix} I \end{bmatrix}$ 8 SensorsSystem Eigenvalues

Before <u>Transformation ($KC_s \neq 0$)</u>		After <u>Transformation ($KC_s = 0$)</u>	
$-.01473 \pm 5.71307i$	-S-	$-.02855 \pm 5.71071i$	
$-.02838 \pm 5.67583i$	-S-	$-.02838 \pm 5.67583i$	
$.02350 \pm 5.15484i$	-S-	$-.02575 \pm 5.14935i$	
$-.05240 \pm 3.84096i$	-S-	$-.01924 \pm 3.84834i$	
$-.01538 \pm 3.55804i$	-S-	$-.01779 \pm 3.55770i$	
$-.02363 \pm 2.96131i$	-S-	$-.01482 \pm 2.96458i$	
$-.07371 \pm 1.46483i$	-C-	$-.07457 \pm 1.46607i$	
$-.05187 \pm 1.17031i$	-C-	$-.05199 \pm 1.17046i$	
$-15.6899 + 0i$	-O-	$-.03632 \pm 1.19306i$	
$-5.60922 + 0i$	-O-	$-.00733 \pm 1.46676i$	
$-1.20084 + 0i$	-O-		
$-1.02196 + 0i$	-O-		

Eigenvalues of $A_C + B_C G$

$-.07457 \pm 1.46607i$	-C-	$-.07457 \pm 1.46607i$
$-.05199 \pm 1.17046i$	-C-	$-.05199 \pm 1.17046i$

Eigenvalues of $A_C - KC_C$

$-15.69921 + 0i$	-O-	$-.03632 \pm 1.19306i$
$-5.62028 + 0i$	-O-	$-.00733 \pm 1.46676i$
$-1.15106 + 0i$	-O-	
$-1.01144 + 0i$	-O-	

Eigenvalues of $A_C + B_C G - KC_C$

$15.33771 + 0i$	$-.07457 \pm 1.46231i$
$4.12626 + 0i$	$-.02153 \pm 1.19748i$
$2.48129 + 0i$	
$1.25725 + 0i$	

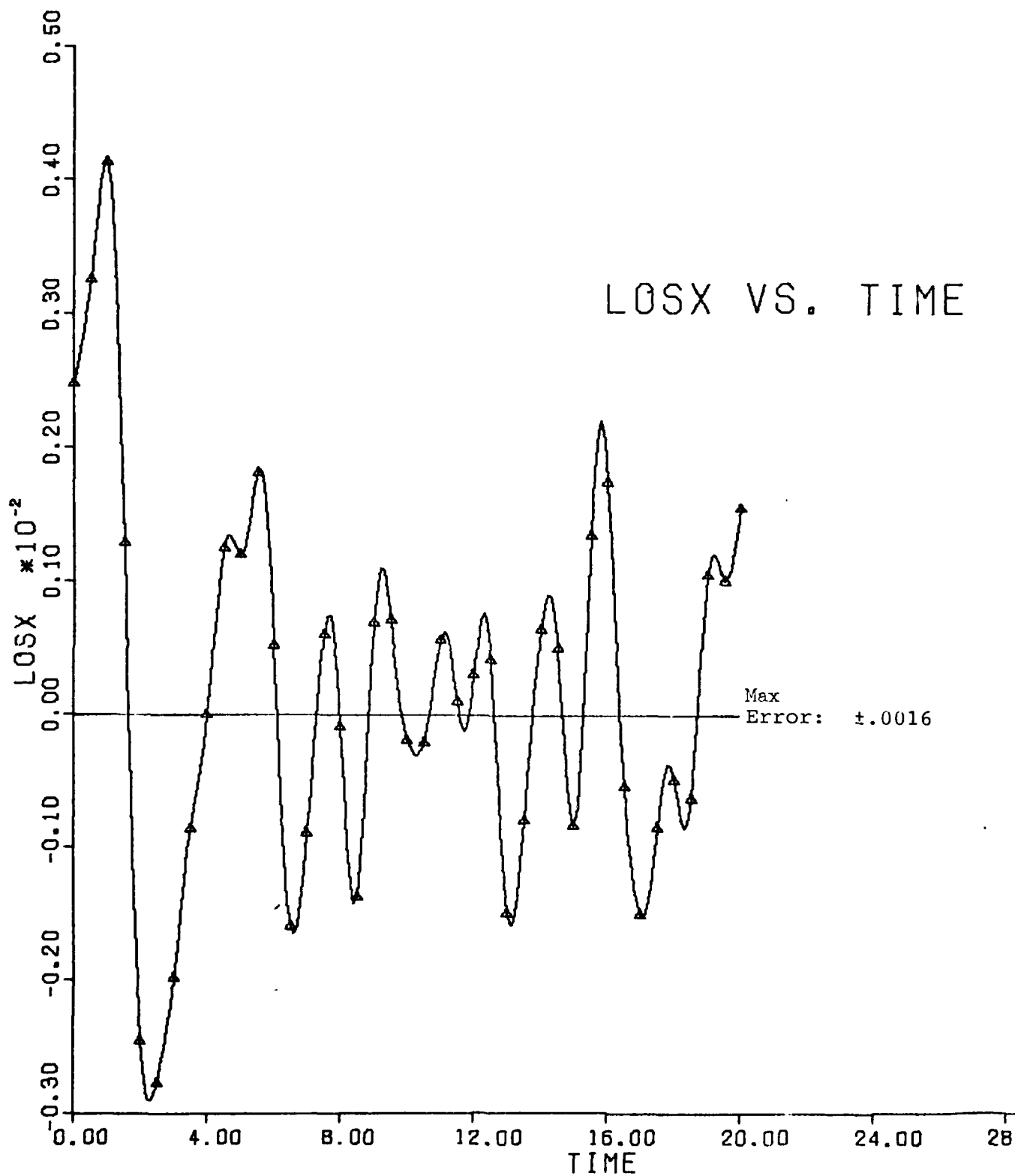


Figure 18. LOSX VS. TIME, $KC_s \neq 0$, 8 Sensors, $Q = 1000.0$

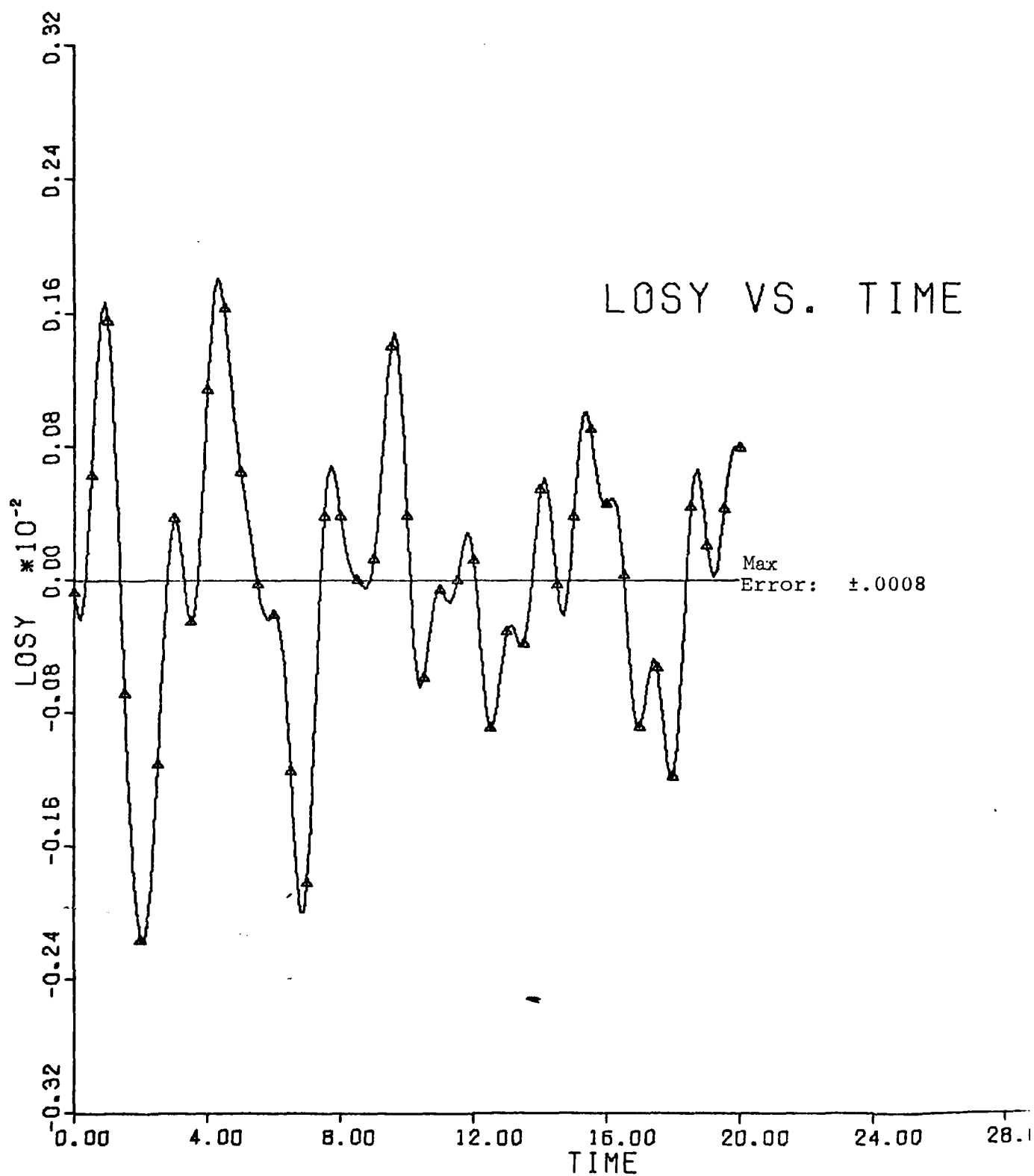


Figure 19. LOS Y VS. TIME, $KC_s \neq 0$, 9 Sensors, $Q = 1000.0$

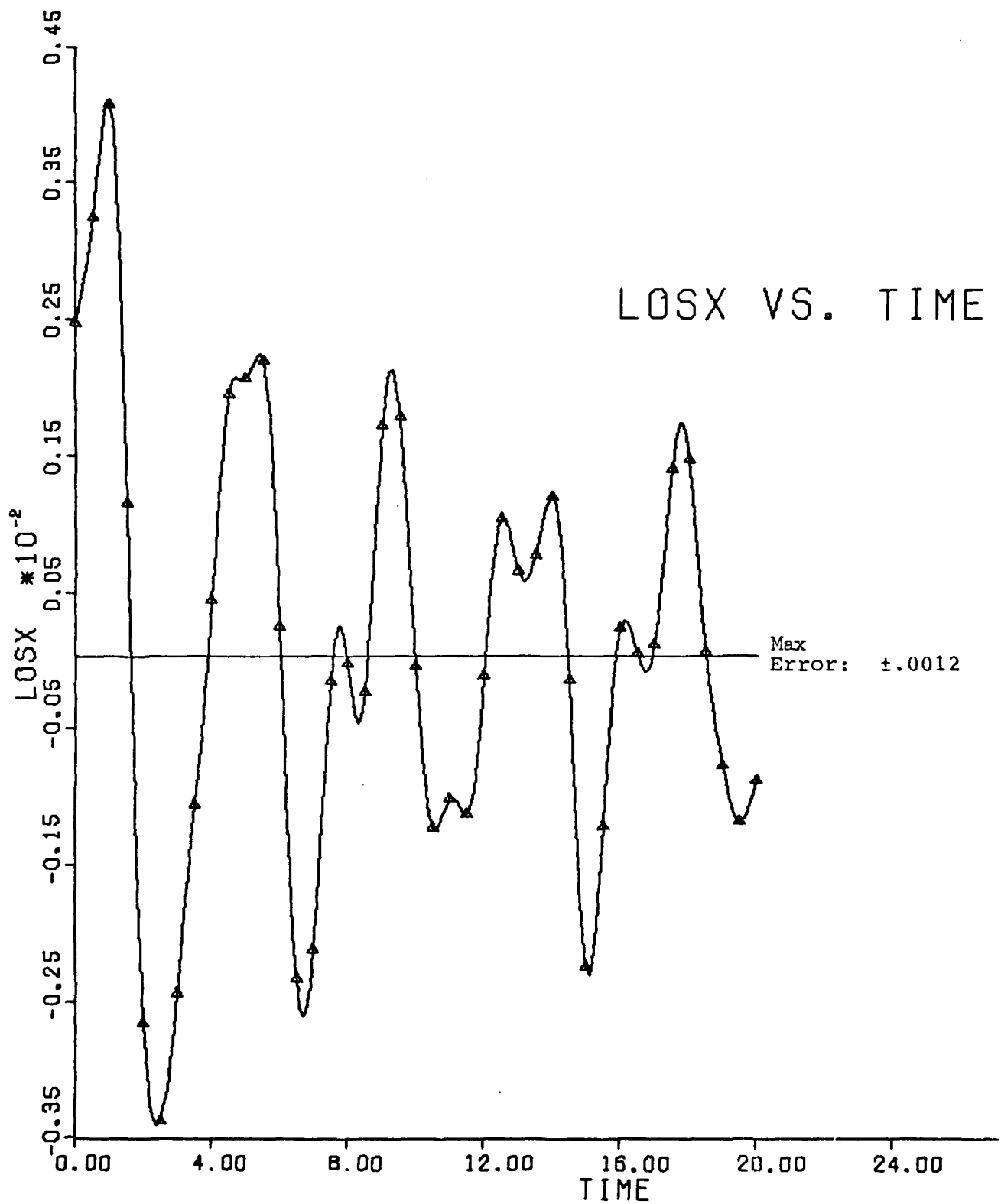


Figure 20. LOSX VS. TIME, $KC_s = 0$, 8 Sensors, $Q = 1000$.

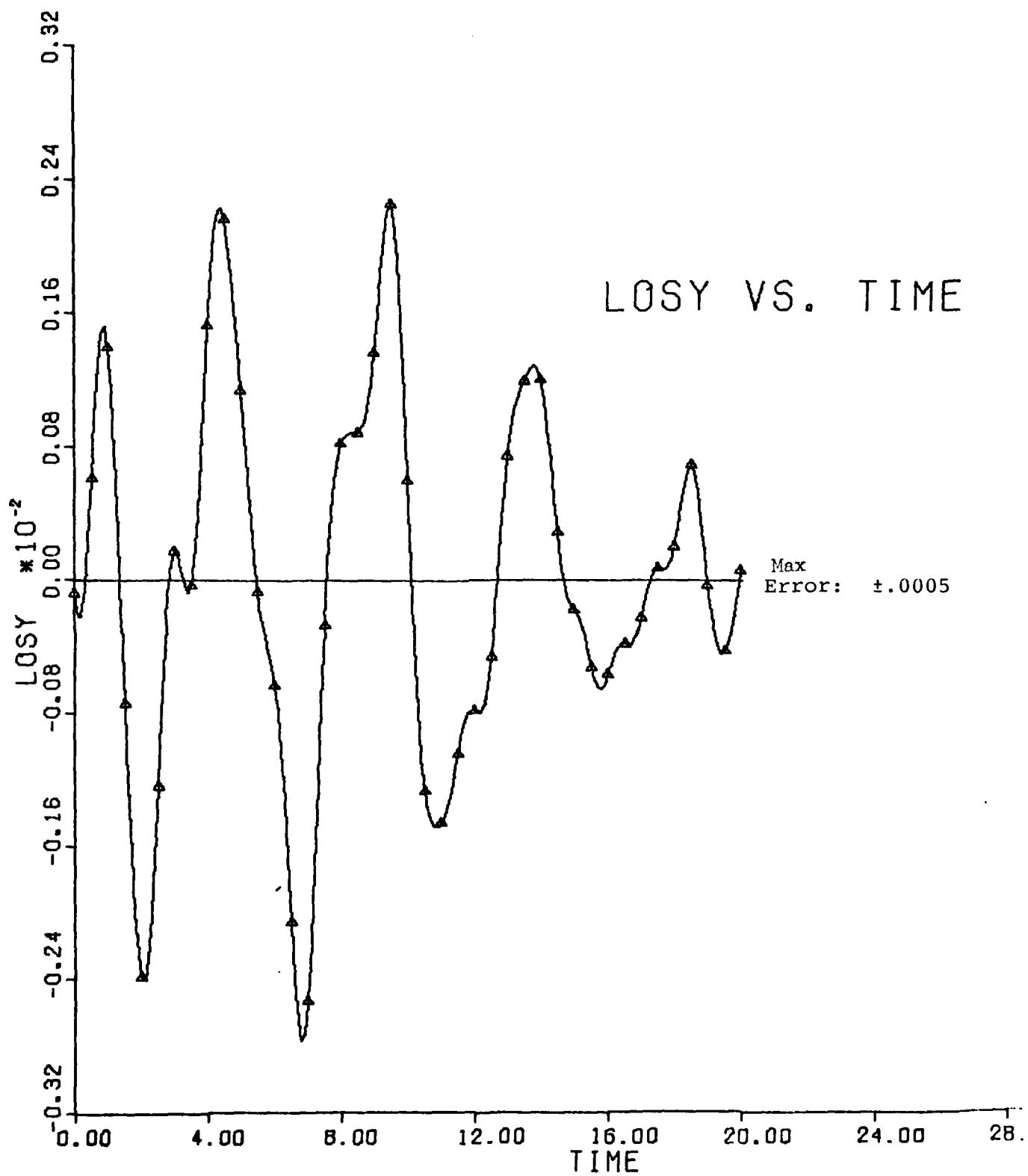


Figure 21. LOS Y VS. TIME, $KC_s = 0$, 8 Sensors, $Q = 1000.0$

Conclusions

Two key conclusions can be drawn from the preceding analyses. First, given a fixed number of sensors and actuators with fixed orientation, the destabilizing effect of observation spillover and control spillover can be "minimized". When a reorientation of those sensors and actuators is permitted, these spillover effects can be completely eliminated. Elimination of either control spillover or observation spillover guarantees system stability, regardless of whether or not response criteria are satisfied. Second, if sensor reorientation is not allowed, complete elimination of observation spillover can still be accomplished through sensor additions.

The transformation method was found to be very effective in eliminating control spillover and uncoupling system eigenvalues when the number of actuators in the system is greater than the number of modes to be suppressed. When the number of modes to be suppressed is equal to the number of actuators, complete elimination of control spillover can be accomplished through an actuator reorientation which reduces the rank of the control matrix, B . A parallel case can be made for the elimination of observation spillover where the number of sensors is greater than or equal to the number of suppressed modes. When reorientation is not permitted, the degree of response improvement is strictly a function of the relative magnitudes of the singular values of the decomposed matrices. For the specific cases examined, the truncation of higher frequency modes was seen to be

valid. This truncation may not hold against other models.

Recommendations

The major theme of this analysis suggests that, due to the complexity of larger and larger space systems, controllers will have to be developed to operate on only those modes critical to system response. This requirement is imposed due to limited computer and hardware capabilities. Since line of sight was established as the performance criteria in this study, the modes were arranged in order of decreasing displacement at the selected grid point. The decision to control two modes and to suppress six was arbitrary. Since the selection of "critical modes" is the starting point in developing an eventual controller, the importance of this step cannot be overemphasized. No automated technique for this process is currently available. An exhaustive re-application of the computer technique found in Appendix A may result in satisfaction of the prescribed time response criteria. More importantly, valuable insight into this task of critical mode selection might be obtained as fallout from this study. In a parallel sense, the selection of two sensors to be added for the final case examined was also arbitrary. Once again, a follow up with varying numbers and locations of additional sensors would be necessary to develop the optimal controller for this model. Finally, sensitivity to modelling inaccuracies would be a natural topic for further analysis. Parameter variations would have to be incorporated into the NASTRAN analysis provided in order to simulate mode shape and frequency errors for this sensitivity study.

Bibliography

1. Balas, M.J. "Active Control of Flexible Systems," AIAA Symposium on Dynamics and Control of Large Flexible Spacecraft, Blacksburg, June 14, 1977.
2. Sesak, J.R. "Control of Large Space Structures Via Singular Perturbation Optimal Control," AIAA Conference On Large Space Platforms: Future Needs and Capabilities, Los Angeles, California, September 27-29, 1978.
3. Coradetti, T. "Orthogonal Subspace Reduction of Optimal Regulator Order," General Dynamics/Convair Division, San Diego, California.
4. Strang, G. Linear Algebra and Its Applications. New York: Academic Press, 1976.
5. Sanborn, K.D. "Modern Optimal Control Methods Applied in Active Control of a Cantilever Beam in Bending Vibration," Unpublished MS Thesis, School of Engineering, Air Force Institute of Technology, Wright Patterson Air Force Base, Ohio, December, 1979.
6. Fager, J.A. "Large Space Erectable Antenna Stiffness Requirements," Journal of Spacecraft and Rockets, 17: 86-92, (March-April, 1980).
7. Meirovitch, L. Methods of Analytical Dynamics. New York: McGraw-Hill Book Company, 1970.
8. Johnson, C.D. "State-Variable Design Methods May Produce Unstable Feedback Controllers," International Journal of Control, 29: 607-619, 1979.
9. Kleinman, D.L. A Description of Computer Programs for Use in Linear Systems Studies. The University of Connecticut School of Engineering TR-77-2. Storrs, Connecticut, July 1977.
10. AFFDL-TM-78-97-FBR. Solution of Ordinary Differential Equations on the CDC 66001/Cyber 74 Processors II, Air Force Flight Dynamics Laboratory, Wright Patterson Air Force Base, Ohio, September, 1978.

Appendix A

Eigenvector Results of NASTRAN Analysis

Real Eigenvectors

Eigenvalue 1 = 1.37043E+00

Eigenvector 1 = $\begin{bmatrix} -2.47073E-01 \\ 4.27857E-02 \\ 1.45180E-06 \\ -1.96263E-02 \\ 3.39753E-02 \\ -7.21326E-02 \\ -3.69602E-02 \\ 4.39747E-02 \\ -1.96224E-02 \\ 5.29624E-02 \\ 4.39672E-02 \end{bmatrix}$

Eigenvalue 3 = 8.78894E+00

Eigenvector 3 = $\begin{bmatrix} 6.36794E-02 \\ 3.67778E-02 \\ 4.00015E-01 \\ 1.98377E-01 \\ 1.14530E-01 \\ 2.00976E-01 \\ 1.54760E-01 \\ 6.80356E-02 \\ 9.78233E-02 \\ 1.36292E-01 \\ 1.00014E-01 \\ 9.78391E-02 \end{bmatrix}$

Eigenvalue 5 = 1.48101E+01

Eigenvector 5 = $\begin{bmatrix} -8.78330E-02 \\ -5.07014E-02 \\ -1.29877E-01 \\ 3.09503E-01 \\ 1.78636E-01 \\ -3.51419E-01 \\ 2.86593E-01 \\ 1.22432E-01 \\ 1.13906E-02 \\ 2.49398E-01 \\ 1.86846E-01 \\ 1.14008E-02 \end{bmatrix}$

Eigenvalue 2 = 2.15145E+00

Eigenvector 2 = $\begin{bmatrix} 3.99896E-01 \\ 2.30929E-01 \\ -1.48908E-01 \\ 8.32862E-02 \\ 4.80849E-02 \\ 6.81283E-02 \\ 6.99996E-02 \\ 2.25294E-02 \\ -4.72104E-02 \\ 5.45051E-02 \\ 4.93610E-02 \\ -4.72153E-02 \end{bmatrix}$

Eigenvalue 4 = 1.26576E+00

Eigenvector 4 = $\begin{bmatrix} 2.74559E-02 \\ -4.75782E-02 \\ -2.24915E-05 \\ -1.71840E-01 \\ 2.97744E-01 \\ -6.81682E-05 \\ -2.51248E-01 \\ 3.43581E-01 \\ -8.19014E-02 \\ -1.71848E-01 \\ 3.89435E-01 \\ 8.19218E-02 \end{bmatrix}$

Eigenvalue 6 = 2.65165E+01

Eigenvector 6 = $\begin{bmatrix} 1.35323E-05 \\ 1.21816E-11 \\ 3.40156E-11 \\ -2.04139E-01 \\ 3.53548E-01 \\ -6.05706E-06 \\ -2.04139E-01 \\ -3.53548E-01 \\ 1.08602E-04 \\ 4.08202E-01 \\ 6.80214E-10 \\ 5.06531E-10 \end{bmatrix}$

Eigenvalue 7 = 3.22159E+01

Eigenvector 7 = $\begin{bmatrix} -2.66140E-02 \\ 4.60655E-02 \\ 3.30215E-05 \\ 3.37411E-02 \\ -5.84417E-02 \\ 3.23144E-05 \\ 2.73330E-02 \\ -5.48104E-02 \\ -4.91269E-01 \\ 3.38171E-02 \\ -5.10814E-02 \\ 4.90852E-01 \end{bmatrix}$

Eigenvalue 9 = 7.99170E+01

Eigenvector 9 = $\begin{bmatrix} 9.90668E-02 \\ 5.72029E-02 \\ 1.72892E-01 \\ 1.07566E-01 \\ 6.21328E-02 \\ -4.95312E-01 \\ -1.67880E-01 \\ -2.19818E-01 \\ -1.11010E-02 \\ -2.74347E-01 \\ -3.55381E-02 \\ -1.10861E-02 \end{bmatrix}$

Eigenvalue 11 = 1.19320E+02

Eigenvector 11 = $\begin{bmatrix} 6.36959E-02 \\ 3.67781E-02 \\ 9.58836E-02 \\ -2.40062E-01 \\ -1.38592E-01 \\ -2.60496E-01 \\ -8.60592E-02 \\ 3.94412E-01 \\ 6.96952E-03 \\ 2.98410E-01 \\ -2.71939E-01 \\ 6.97073E-03 \end{bmatrix}$

Eigenvalue 8 = 3.26133E+01

Eigenvector 8 = $\begin{bmatrix} -2.99367E-02 \\ -1.73093E-02 \\ 8.78423E-02 \\ 4.07052E-02 \\ 2.35996E-02 \\ 3.55373E-02 \\ 2.74211E-02 \\ 2.79794E-02 \\ -4.87453E-01 \\ 3.79914E-02 \\ 9.80954E-03 \\ -4.87867E-01 \end{bmatrix}$

Eigenvalue 10 = 1.06164E+02

Eigenvector 10 = $\begin{bmatrix} -3.38986E-03 \\ 5.84999E-03 \\ -1.60534E-05 \\ -2.28617E-01 \\ 3.95968E-01 \\ 4.96376E-05 \\ 3.78349E-01 \\ 4.55436E-02 \\ -1.47053E-02 \\ -2.28600E-01 \\ -3.04859E-01 \\ 1.47172E-02 \end{bmatrix}$

Eigenvalue 12 = 1.95068E+02

Eigenvector 12 = $\begin{bmatrix} 3.20580E-02 \\ 1.85105E-02 \\ 6.43806E-02 \\ -4.02579E-01 \\ -2.32435E-01 \\ -1.30450E-01 \\ 3.20382E-01 \\ -1.58741E-01 \\ -9.27787E-03 \\ 2.27168E-02 \\ 3.56828E-01 \\ -9.28169E-03 \end{bmatrix}$

Eigenvector = $\{x_1, y_1, z_1, \dots, z_4\}^T$

Appendix B
Main Program Listing


```

1  PROGRAM TETRA(INPUT=/80,OUTPUT=/132,TAPE5,TAPE6,TAPE7)
   DIMENSION Y(20),YP(20),WORK(520),IWORK(5)
   DIMENSION CCN8(8,16),CSP8(8,16),CCNT(16,8),CT(16,1)
   DIMENSION T1(1,6),TT1(6,1),V2(16,1),CTT(1,16)
5  DIMENSION VV2(16,16),S2(16,16),ULX2(16,15),OK1(1,16)
   DIMENSION OK2(1,16),OKT(16,15)
   DIMENSION XX1(203),XX2(203),XX3(203),XX4(203),XX5(203),XX6(203)
   DIMENSION XX7(203),XX8(203),TM(203),X1(203),X2(203)
10  DIMENSION Z8(20,20)
   DIMENSION T2(2,8),TT2(8,2),CT2(15,2),RT2(2,8),RT3(2,2)
   DIMENSION V3(4,2),CTT2(2,16),VV3(4,4),S3(16,16),ULX3(16,15)
   DIMENSION RTII(2,2),OK3(2,15),OK4(2,16),OKT2(8,4)
   DIMENSION BO(16,16),SV(8),WL(16)
   DIMENSION R1(8,8),RR1(8,8),RRR1(8,8)
15  DIMENSION PH1(12,16),PHIT(15,12),J(12,6),J1(8,12)
   DIMENSION PHIT5(8,12),PHI5(12,8),BK(8,6),BK6(8,4),CR(8,8)
   DIMENSION KCR(16,8)
   DIMENSION CSP4(12,8)
20  DIMENSION ATOT(16,16),BTOT(15,6),CTOT(8,16)
   DIMENSION ACON(16,16),ASUP(16,15),JCON(15,6),BSUP(16,6)
   DIMENSION CSP9(16,8)
   DIMENSION CY(16,16),ACN6(16,16)
   DIMENSION WK(32),CCT(16,16),CN(15,16)
   DIMENSION WK1(32),WK2(32),WK3(32)
25  DIMENSION ACN1(16,16),UC(16,16)
   DIMENSION CCON(8,16),CSUP(8,16)
   DIMENSION CCN2(8,16),CCN1(15,8),BCN1(16,5)
   DIMENSION ACN2(16,16),CX(16,16),ULX(16,15)
   DIMENSION BCN2(6,16),CSP1(8,16),BSP1(16,5)
30  DIMENSION COM3(16,16),C(16,16)
   DIMENSION R(6,6),Q(16,16),RY(6,15),ACONT(16,16),CCONT(16,8)
   DIMENSION RX(8,16),P(16,16),OKTRN(3,16),OK(16,8),RR(6,6)
   DIMENSION RTT(1,16),UL(16,15),CC(15,16),OSPILL(16,16)
   DIMENSION CSPILL(16,16),FJ(16,16)
35  DIMENSION COM1(16,16),COM2(16,15),S(16,15),GO(6,16)
   DIMENSION CO(16,16),COC(16,16)
   DIMENSION SS(16,16),RRK(6,6),RU(15,8),RV(16,6)
   DIMENSION Z7(15,16),SO(16,15),BCONT(5,15)
   DIMENSION PHIT1(16,12),BSP3(16,5),BSP4(15,6),PHIT2(16,12)
40  DIMENSION T(6,1),TT(1,6),RT(16,1),RT1(1,5)
   DIMENSION RT(1,1),V1(16,1),VV(16,15),S1(16,16)
   DIMENSION ULX1(16,16),QQ(15,16),RTI(1,1),G01(1,15),G02(1,16)
   COMPLEX W(20),Z0(20,20),WW(5),Z00(8,8)
   COMPLEX W1(6),W2(8),W3(8),Z1(8,8),Z2(8,8),Z3(6,8)
45  INTEGER FFF,F1,F2,F3,F4,GGG,G1,G2,G3,G4
   INTEGER N5,N6,N7
   INTEGER KZ
   INTEGER NB,KK
   INTEGER F5,N7
50  INTEGER IFLAG,NEON,JJ
   INTEGER NY
   INTEGER INIT
   INTEGER IZ,NN,IJOB
   INTEGER L,MO,NO,IA,IB,IC
55  INTEGER I,J,N,M,F,G,FF,GG
   INTEGER NDIM,NDIM1,KIN,KOUT,KPUNC4
   REAL TI,TOUT,RELERR,ABSERR

```

```

      REAL C1,Q2,Q3,Q4,Q5
      REAL Q6
60     REAL TOL
      REAL CAMP
      REAL EB
      COMMON Z9(20,20)
      COMMON/MAIN1/NDIM,NDIM1,COM1
65     COMMON/INOUT/KOUT,KIN,KPUNCH
      COMMON/MAIN2/COM2
      COMMON/MAIN3/COM3
      EXTERNAL XDOT
      READ*,N5
70     NDIM=16
      NDIM1=17
      TOL=.001
      KIN=5
      KOUT=6
75     KPUNCH=7
      IER=0
      DAMP=.005
      NY=0
      N7=0
80     KK=0
      M=8-N5
      F5=2*N5
      G=2*M
      FF=N5+1
85     GG=M+1
      F1=F5+1
      F2=2*F5
      F3=F2+1
      F4=F2+G
90     IF(KK.GT.0)GO TO 37
      READ*,N6
      READ*,N7

      DO 20 I=1,16
      DO 10 J=1,16
100     ATOT(I,J)=0.0
      CONTINUE
      CONTINUE
      DO 30 I=1,8
      J=I+8
105     ATOT(I,J)=1.0
      CONTINUE
      DO 35 I=9,16
      J=I-8
      READ*,ATOT(I,J)
110     CONTINUE
      DO 36 I=9,16
      READ*,ATOT(I,I)
      ATOT(I,I)=-DAMP*2.*ATOT(I,I)
35     CONTINUE

```

```

115      37      PRINT*, " "
          PRINT*, " "
          PRINT*, " "
          PRINT*, "THIS RUN REPRESENTS AN ANALYSIS FOR AN EIGHT MODE"
          PRINT*, "APPROXIMATION TO THE SYSTEM, WITH ", N5, " MODES"
120      PRINT*, "CONTROLLED AND ", M, " MODES SUPPRESSED"
          PRINT*, " "
          IF (NZ.GT.0) GO TO 169
          PRINT*, " "

          0
          0
125      0      CREATE B MATRIX
          0
          0
          32      DO 70 I=1,8
130      DO 50 J=1,6
          BTOT(I,J)=0.0
          50      CONTINUE
          70      CONTINUE
          DO 72 I=1,16
135      DO 71 J=1,16
          C(I,J)=0.0
          IF (I.EQ.J) C(I,J)=1.0
          71      CONTINUE
          72      CONTINUE
140      DO 74 I=1,12
          DO 73 J=1,6
          D(I,J)=0.0
          73      CONTINUE
          74      CONTINUE
145      DO 76 I=4,6
          DO 75 J=1,2
          READ*, D(I,J)
          75      CONTINUE
          75      CONTINUE
150      DO 78 I=7,9
          DO 77 J=3,4
          READ*, D(I,J)
          77      CONTINUE
          78      CONTINUE
155      DO 80 I=10,12
          DO 79 J=5,6
          READ*, D(I,J)
          79      CONTINUE
          80      CONTINUE
160      DO 82 I=1,12
          DO 81 J=1,8
          PHI(I,J)=0.0
          81      CONTINUE
          82      CONTINUE
165      DO 84 J=9,16
          DO 83 I=1,12
          READ*, PHI(I,J)
          83      CONTINUE
          84      CONTINUE
170      DO 86 I=1,16
          DO 85 J=1,12

```

```

      PHIT(I,J)=PHI(J,I)
      85 CONTINUE
      95 CONTINUE
175  97 L1=16
      MO=12
      NO=6
      IA=15
      IB=12
180  IC=16
      CALL VMULFF(PHIT,D,L1,MO,NO,IA,IB,CTOT,IC,IER)
      C
      C
      C CREATE C MATRIX
185  C
      C
      IF(NY.GT.6)GO TO 58
121  IF(N6.GT.6)GO TO 113
      DO 111 I=1,6
190  DO 110 J=1,12
      D1(I,J)=D(J,I)
110 CONTINUE
111 CONTINUE
118 L1=N6
195  MO=12
      NO=16
      IA=8
      IB=12
      IC=8
200  CALL VMULFF(D1,PHI,L1,MO,NO,IA,IB,CTOT,IC,IER)
      DO 146 I=1,N6
      DO 145 J=1,8
      CTOT(I,J)=CTOT(I,8+J)
145 CONTINUE
205 146 CONTINUE
      DO 148 I=1,N6
      DO 147 J=9,16
      CTOT(I,J)=0.0
147 CONTINUE
210 148 CONTINUE
      GO TO 150
113 DO 115 I=1,2
      DO 114 J=1,12
      D1(I,J)=0.0
215 114 CONTINUE
115 CONTINUE
      D1(1,1)=.3535533906
      D1(1,2)=-.6122837618
      D1(1,3)=.7071067812
22  D1(2,1)=-.3535533906
      D1(2,2)=.6122837618
      D1(2,3)=.7071067812
      DO 117 I=3,8
      DO 116 J=1,12
      D1(I,J)=D(J,I-2)
      116 CONTINUE
      117 CONTINUE
      L1=N6

```

```
230      MO=12
        NO=16
        IA=9
        IB=12
        IC=8
        CALL VMULFF(D1,PHI,L1,MO,NO,IA,IB,CTOT,IC,IER)
235      DO 54 I=1,8
        DO 53 J=1,8
        CTOT(I,J)=CTOT(I,J+8)
53      CONTINUE
54      CONTINUE
240      DO 56 I=1,8
        DO 55 J=9,16
        CTOT(I,J)=0.0
55      CONTINUE
56      CONTINUE
245      150 IF(N7.GT.0)GO TO 134
        ?
        0
        0
        0
        0
250      160 DO 180 I=1,N5
        DO 170 J=1,N5
        ACON(I,J)=ATOT(I,J)
170     CONTINUE
255     180 CONTINUE
        DO 200 I=1,N5
        DO 190 J=FF,F5
        ACON(I,J)=ATOT(I,M+J)
190     CONTINUE
260     200 CONTINUE
        DO 220 I=FF,F5
        DO 210 J=1,N5
        ACON(I,J)=ATOT(I+M,J)
210     CONTINUE
265     220 CONTINUE
        DO 240 I=FF,F5
        DO 230 J=FF,F5
        ACON(I,J)=ATOT(I+M,J+M)
230     CONTINUE
270     240 CONTINUE
54      DO 260 I=1,M
        DO 250 J=1,M
        ASUP(I,J)=ATOT(I+N5,J+N5)
250     CONTINUE
275     260 CONTINUE
        DO 280 I=1,M
        DO 270 J=GG,G
        ASUP(I,J)=ATOT(I+N5,J+2*N5)
270     CONTINUE
280     280 CONTINUE
        DO 300 I=GG,G
        DO 290 J=1,M
        ASUP(I,J)=ATOT(2*N5+I,N5+J)
290     CONTINUE
285     300 CONTINUE
```

```
      DO 320 I=GG,G
      DO 310 J=GG,G
      ASUP(I,J)=ATOT(2*N5+I,2*N5+J)
290      310 CONTINUE
      320 CONTINUE
      C
      C
      C CREATE B CONTROLLED AND B SUPPRESSED
      C
295      330 DO 340 I=1,N5
      DO 330 J=1,6
      BCON(I,J)=BTOT(I,J)
      330 CONTINUE
300      340 CONTINUE
      DO 360 I=FF,F5
      DO 350 J=1,6
      BCON(I,J)=BTOT(I+M,J)
      350 CONTINUE
305      360 CONTINUE
      173 DO 380 I=1,M
      DO 370 J=1,6
      BSUP(I,J)=BTOT(N5+I,J)
      370 CONTINUE
310      380 CONTINUE
      DO 400 I=GG,G
      DO 390 J=1,6
      BSUP(I,J)=BTOT(2*N5+1,J)
      390 CONTINUE
315      400 CONTINUE
      IF(NY.GT.0)GO TO 561
      C
      C
      C CREATE C CONTROLLED AND C SUPPRESSED
      C
320      104 DO 420 I=1,N6
      DO 410 J=1,N5
      CCON(I,J)=CTOT(I,J)
325      410 CONTINUE
      420 CONTINUE
      DO 440 I=1,N6
      DO 430 J=FF,F5
      CCON(I,J)=0.0
330      430 CONTINUE
      440 CONTINUE
      162 DO 460 I=1,N6
      DO 450 J=1,M
      CSUP(I,J)=CTOT(I,N5+J)
335      450 CONTINUE
      460 CONTINUE
      DO 480 I=1,N6
      DO 470 J=GG,G
      CSUP(I,J)=CTOT(I,2*N5+J)
340      470 CONTINUE
      480 CONTINUE
      C
```

```

345      3
        3
        3
        3
        3      CREATE WEIGHTING MATRICES
165      IF(N7.GT.0)GO TO 541
        DO 491 I=1,6
        DO 490 J=1,6
350      R(I,J)=0.0
        IF(I.EQ.J)R(I,J)=1.0
491      CONTINUE
491      CONTINUE
        IF(N6.GT.6)GO TO 501
355      R3=1.
        DO 493 I=1,6
        DO 492 J=1,6
        R1(I,J)=0.0
        IF(I.EQ.J)R1(I,J)=1.0
360      492      CONTINUE
        493      CONTINUE
        GO TO 502
501      DO 497 I=1,8
        DO 496 J=1,8
365      R1(I,J)=0.0
        IF(I.EQ.J)R1(I,J)=1.0
        496      CONTINUE
        497      CONTINUE
        R3=1.
370      502      DO 520 I=1,F5
        DO 510 J=1,F5
        Q(I,J)=0.0
        IF(I.EQ.J)Q(I,J)=88
        510      CONTINUE
375      520      CONTINUE
        522      DO 540 I=1,F5
        DO 530 J=1,F5
        ACONT(I,J)=ACON(J,I)
        530      CONTINUE
380      540      CONTINUE
        541      DO 560 I=1,F5
        DO 550 J=1,N6
        CCONT(I,J)=CCON(J,I)
        550      CONTINUE
385      560      CONTINUE
        IF(N7.GT.0)GO TO 625
        561      DO 580 I=1,6
        DO 570 J=1,F5
        BCONT(I,J)=BCON(J,I)
390      570      CONTINUE
        580      CONTINUE
        PRINT*,"GO"
        IF(NY.GT.0)GO TO 562
        DO 601 I=1,6
        DO 600 J=1,6
395      RR(I,J)=1.0
        600      CONTINUE
        601      CONTINUE
        DO 610 I=1,6

```

PROGRAM TETRA

```

401      RR(I,I)=-1./R(I,I)
        510      CONTINUE
              DO 512 I=1,6
              DO 511 J=1,6
              RRR(I,J)=-RF(I,J)
405      511      CONTINUE
        512      CONTINUE
              IF(N6.GT.6)GO TO 619
              DO 514 I=1,6
              DO 513 J=1,6
410      RR1(I,J)=0.0
        513      CONTINUE
        514      CONTINUE
              DO 515 I=1,6
              RR1(J,I)=-1./R(I,I)
415      515      CONTINUE
              DO 518 I=1,6
              DO 517 J=1,6
              RRR1(I,J)=-FR1(I,J)
        517      CONTINUE
420      518      CONTINUE
              GO TO 525
        519      DO 521 I=1,6
              DO 520 J=1,6
              RR1(I,J)=0.0
425      520      CONTINUE
        521      CONTINUE
              DO 522 I=1,6
              RR1(J,I)=-1./R1(I,I)
        522      CONTINUE
430      DO 524 I=1,6
              DO 523 J=1,6
              RRR1(I,J)=-RR1(I,J)
        523      CONTINUE
        524      CONTINUE
435      3
        3
        3      SOLVE RESPECTIVE RICCATI EQUATIONS
        3
        3
440      525      L1=F5
              MO=N6
              NO=N6
              IA=16
              IB=5
445      IC=16
              CALL VMULFF(CCONT,RRR1,L1,MO,NO,IA,IB,RU,IC,IER)
              NO=F5
              CALL VMULFF(RU,CCON,L1,MO,NO,IA,IB,SS,IC,IER)
              IF(N7.GT.0)GO TO 637
450      526      DO 528 I=1,F5
              DO 527 J=1,F5
              ACN1(I,J)=ACON(I,J)
              ACN2(I,J)=ACONT(I,J)
        527      CONTINUE
455      528      CONTINUE
        529      DO 530 I=1,F5

```



```

DO 529 J=1,6
  ACN1(I,J)=BCON(I,J)
  BCN2(J,I)=BCONT(J,I)
460      529 CONTINUE
      630 CONTINUE
DO 632 I=1,6
DO 631 J=1,6
  BSP1(I,J)=BSUP(I,J)
465      IF(NZ.GT.0)BSP1(1,J)=BSP4(I,J)
      531 CONTINUE
      632 CONTINUE
  IF(NY.GT.0)GO TO 563
DO 634 I=1,F5
470      DO 533 J=1,N6
  CCN1(I,J)=CCONT(I,J)
  CCN2(J,I)=CCON(J,I)
      533 CONTINUE
      634 CONTINUE
DO 536 I=1,N6
475      DO 635 J=1,G
  CSP1(I,J)=CSUP(I,J)
      535 CONTINUE
      536 CONTINUE
480      637 CALL MRIC(F5,ACN2,SS,Q,P,OX,TOL,IER)
  L1=N6
  MO=N6
  NO=F5
  IA=8
  IR=8
  IC=8
  CALL VMULFF(RR1,CCN2,L1,MO,NO,IA,IR,RX,IC,IER)
  MO=F5
  IR=16
490      CALL VMULFF(RX,P,L1,MO,NO,IA,IR,OKTRN,IC,IER)
DO 550 I=1,F5
DO 540 J=1,N6
  OK(I,J)=-OKTRN(J,I)
      540 CONTINUE
495      550 CONTINUE
  IF(NZ.GT.0)GO TO 671
      551 DO 570 I=1,F5
DO 660 J=1,F5
  FO(I,J)=0.0
500      IF(I.EQ.J)FO(I,J)=1.0
      550 CONTINUE
      570 CONTINUE
      563 L1=F5
  MO=6
505      NO=6
  IA=16
  IR=6
  IC=16
  CALL VMULFF(BCON,RRR,L1,MO,NO,IA,IR,RV,IC,IER)
  NO=F5
510      CALL VMULFF(RV,BCONT,L1,MO,NO,IA,IR,SO,IC,IER)
  CALL MRIC(F5,ACN1,SO,FO,S,ULX,TOL,IER)
  L1=6

```

```

515      MO=5
      IA=6
      IR=5
      IC=5
      NO=F5
      CALL VMULFF(RR,BCN2,L1,MO,NO,IA,IB,RY,IC,IER)
520      MO=F5
      IR=16
      CALL VMULFF(RY,S,L1,MO,NO,IA,IB,GO,IC,IER)
571      L1=F5
      MO=5
      NO=F5
525      IA=16
      IR=6
      IC=16
      CALL VMULFF(BCN1,GO,L1,MO,NO,IA,IB,UC,IC,IER)
530      MO=N6
      IR=9
      IA=16
      CALL VMULFF(OK,CCN2,L1,MO,NO,IA,IB,CC,IC,IER)
535      DO 592 I=1,F5
      DO 591 J=1,F5
      CCT(I,J)=CC(J,I)
591      CONTINUE
592      CONTINUE
      DO 594 I=FF,F5
      DO 593 J=1,N5
540      CN(I,J)=CCT(I-N5,J+N5)
593      CONTINUE
594      CONTINUE
      DO 596 I=FF,F5
      DO 595 J=FF,F5
545      CN(I,J)=CCT(I-N5,J-N5)
595      CONTINUE
596      CONTINUE
      DO 598 I=1,N5
      DO 597 J=1,F5
550      CN(I,J)=0.0
597      CONTINUE
598      CONTINUE
      DO 730 I=1,F5
      DO 720 J=1,F5
555      CO(I,J)=CN(I,J)

      CC(I,J)=ACN1(I,J)-CN(I,J)
      UL(I,J)=ACN1(I,J)+UC(I,J)
560      COC(I,J)=ULX(I,J)-CX(I,J)+ACN2(I,J)
720      CONTINUE
730      CONTINUE
      L1=F5
      MO=N6
565      NO=G
      IA=16
      IR=9
      IC=16
      CALL VMULFF(OK,CSP1,L1,MO,NO,IA,IB,OSPILL,IC,IER)
570      L1=G

```



```

DO 911 J=1,20
78(I,J)=Z9(I,J)
630      911 CONTINUE
        912 CONTINUE
        998 IJOB=0
          NN=2*F5+G
          IA=20
          I7=20
        C
        C
        C EIGENVALUE ANALYSIS
        C
640      CALL EIGRF(Z9,NN,IA,IJOB,W,ZC,I7,WK,IER)
        DO 914 I=1,20
          DO 913 J=1,20
            79(I,J)=Z8(I,J)
645      913 CONTINUE
        914 CONTINUE
          PRINT*, " "
          PRINT*, "Q=", BB
          PRINT*, " "
650      PRINT*, "THE SYSTEM EIGENVALUES"
          PRINT*, " "
          PRINT*, " "
          PRINT*, " ", "EIGENVALUE"
          PRINT*, " ", "-----"
655      NN=NN-1
          DO 930 I=1,NN,2
            PRINT*, " "
            PRINT*, " ", W(I)
            J=I+1
            PRINT*, " ", A(J)
660      930 CONTINUE
            IA=16
            I7=8
            CALL EIGRF(ULX,F5,IA,IJOB,W1,700,I7,WK,IER)
            PRINT*, " "
            PRINT*, " "
            PRINT*, "THE EIGENVALUES"
            PRINT*, "OF A+B3", " ", "EIGENVALUE"
            PRINT*, " ", "-----"
670      F5=F5-1
            DO 931 I=1,F5,2
              PRINT*, " "
              PRINT*, " ", WH(I)
              J=I+1
              PRINT*, " ", WH(J)
675      931 CONTINUE
              F5=F5+1
              DO 933 I=1,F5
                DO 932 J=1,F5
                  CY(I,J)=CX(I,J)
680      932 CONTINUE
              933 CONTINUE
              CALL EIGRF(CY,F5,IA,IJOB,W1,71,I7,WK1,IER)
              DO 935 I=1,F5

```

```

685      DO 934 J=1,F5
          CX(I,J)=CY(I,J)
          934 CONTINUE
          935 CONTINUE
          PRINT*," "
690      PRINT*," "
          PRINT*,"THE EIGENVALUES"
          PRINT*,"OF A-KC","          ","EIGENVALUE"
          PRINT*,"          ","-----"
          F5=F5-1
695      DO 952 I=1,F5,2
          PRINT*," "
          PRINT*,"          ",W1(I)
          J=I+1
          PRINT*,"          ",W1(J)
700      952 CONTINUE
          F5=F5+1
          CALL EIGRF(COC,F5,IA,IJOB,W2,Z2,I7,WK2,IER)
          PRINT*," "
          PRINT*," "
705      PRINT*,"THE EIGENVALUES"
          PRINT*,"OF A+BG-KC","          ","EIGENVALUE"
          PRINT*,"          ","-----"
          F5=F5-1
          DO 955 I=1,F5,2
710      PRINT*," "
          PRINT*,"          ",W2(I)
          J=I+1
          PRINT*,"          ",W2(J)
          955 CONTINUE
          F5=F5+1
715      DO 968 I=1,F5
          DO 967 J=1,F5
          ACN6(I,J)=ACN1(I,J)
          967 CONTINUE
          968 CONTINUE
720      CALL EIGRF(ACN1,F5,IA,IJOB,W3,Z3,I7,WK3,IER)
          DO 977 I=2,F5
          DO 969 J=1,F5
          ACN2(I,J)=ACN6(I,J)
725      969 CONTINUE
          970 CONTINUE
          PRINT*," "
          PRINT*," "
          PRINT*,"THE EIGENVALUES"
730      PRINT*,"OF A          EIGENVALUE"
          PRINT*,"          -----"
          F5=F5-1
          DO 956 I=1,F5,2
          PRINT*," "
735      PRINT*,"          ",W3(I)
          J=I+1
          PRINT*,"          ",W3(J)
          956 CONTINUE
          F5=F5+1
740      2
          2

```

```

C      INITIAL CONDITIONS
C
C

```

```

745      Y(1)=-.001
          Y(2)=.006
          Y(3)=-.003
          Y(4)=.01
          Y(5)=-.001
750      Y(6)=.006
          Y(7)=-.003
          Y(8)=.01
          Y(9)=.001
          Y(10)=-.009
755      Y(11)=.006
          Y(12)=-.001
          Y(13)=-.002
          Y(14)=.002
          Y(15)=.03
760      Y(16)=-.02
          Y(17)=.02
          Y(18)=-.02
          Y(19)=-.003
          Y(20)=.004
765

```

```

C
C
C      INTEGRATE STATE EQUATIONS
C
C

```

```

770      TI=0.0
          TOUT=.1
          IFLAG=1
          NEQN=20
          ABSERR=1.0E-03
775      RELERR=1.0E-03
          JJ=1
1102     CALL ODE(XDOT,NEQN,Y,TI,TOUT,RELERR,ABSERR,IFLAG,WORK,IWORK)
          IF(TI.LT.TOUT)GO TO 1102
          XX1(JJ)=Y(1)
780      XX2(JJ)=Y(2)
          XX3(JJ)=Y(9)
          XX4(JJ)=Y(10)
          XX5(JJ)=Y(11)
          XX6(JJ)=Y(12)
785      XX7(JJ)=Y(13)
          XX8(JJ)=Y(14)
          JJ=JJ+1
          TOUT=TOUT+.1
          IF(TI.LE.20.)GO TO 1102

```

```

790      C
          C
          C      GENERATE AND PLOT LINE OF SIGHT X AND Y
          C
          C

```

```

795      DO 1103 I=1,201
          TM(I)=(I-1)*.1
1103     CONTINUE
          DO 1104 I=1,201

```

```

      X1(I)=-.2470727*XX1(I)+.3998955*XX2(I)
800  X1(I)=X1(I)-.08783301*XX3(I)+.05357944*XX4(I)
      X1(I)=X1(I)+.02745586*XX5(I)-.02661401*XX6(I)
      X1(I)=X1(I)-.02993670*XX7(I)+.00001353225*XX8(I)
      X2(I)=.4278569*XX1(I)+.2309291*XX2(I)
      X2(I)=X2(I)-.05070142*XX3(I)+.03577784*XX4(I)
805  X2(I)=X2(I)-.04757822*XX5(I)+.04605553*XX6(I)
      X2(I)=X2(I)-.01730931*XX7(I)+.0000000001218161*XX8(I)
110  CONTINUE
      CALL PLOT(0.,0.,-3)
      CALL SCALE(TM,8.,201,1)
810  CALL SCALE(X1,8.,201,1)
1101 CALL AXIS(0.,0.,4HTIME,-4,8.,0.,TM(202),TM(203))
      CALL AXIS(0.,0.,4HLOX,4,8.,90.,X1(202),X1(203))
      CALL LINE(TM,X1,201,1,5,2)
      CALL SYMBOL(4.,6.,0.21,13HLOX VS. TIME,0.,13)
815  CALL PLOT(10.,0.,-3)
      CALL SCALE(TM,8.,201,1)
      CALL SCALE(X2,8.,201,1)
1102 CALL AXIS(0.,0.,4HTIME,-4,8.,0.,TM(202),TM(203))
      CALL AXIS(0.,0.,4HLOS,4,8.,90.,X2(202),X2(203))
820  CALL LINE(TM,X2,201,1,5,2)
      CALL SYMBOL(4.,6.,0.21,13HLOS VS. TIME,0.,13)
      CALL PLOT(N)
      IF(M.LT.N7)GO TO 1101
      IF(KK.GT.0)GO TO 1101
825  Q5=2.0
      Q5=.001
      DO 959 I=1,M
      DO 958 J=1,12
      PHIT1(I,J)=0.0
830  958 CONTINUE
      959 CONTINUE
      DO 961 I=GG,G
      DO 960 J=1,12
      PHIT1(I,J)=PHIT(I+F5,J)
835  960 CONTINUE
      961 CONTINUE
      962 L1=G
      MO=1.2
      NO=5
840  IA=16
      IB=12
      IC=15
      CALL VMULFF(PHIT1,D,L1,MO,NO,IA,IB,BSP3,IC,IER)
      DO 964 I=1,G
      DO 963 J=1,6
      BSP4(I,J)=BSP3(I,J)
      963 CONTINUE
      964 CONTINUE
      DO 966 I=1,G
      DO 965 J=1,G
850  C(I,J)=0.0
      IF(I.EQ.J)C(I,J)=1.0
      965 CONTINUE
      966 CONTINUE
855  IA=16

```

```

      MO=5
      NO=6
      IC=16
      NB=6
860      C
      C
      C
      C
      C
      CHECK FOR ZERO SINGULAR VALUE
865      CALL LSVDF(BSP3,IA,MO,NO,C,IC,NB,S,WK,IER)
      Q4=S(6)
      IF(Q4.LT.05)GO TO 1075
      Q5=Q6 -.61
      IF(Q6.LT.-2.0)GO TO 1096
870      Q1=(4.0+Q6**2.)**.5
      D(7,3)=1./Q1
      D(8,3)=1.7318/Q1
      D(9,3)=Q5/Q1
      GO TO 962
875      1075 PRINT*," "
      PRINT*,"THE LEAST SINGULAR VALUE IS ",S(5)
      PRINT*,"Q6 = ",Q6
      C
      C
      C
      C
880      APPLY TRANSFORMATION TECHNIQUE
      DO 1076 J=1,6
      TT(1,J)=BSP3(6,J)
885      1076 CONTINUE
      DO 1078 I=1,6
      T(I,1)=TT(1,I)
      1078 CONTINUE
      DO 1080 I=1,N5
      DO 1079 J=1,12
      PHIT2(I,J)=0.0
890      1079 CONTINUE
      1080 CONTINUE
      DO 1082 I=FF,F5
      DO 1081 J=1,12
      PHIT2(I,J)=PHIT(I+M,J)
895      1081 CONTINUE
      1082 CONTINUE
      L1=F5
      MO=12
      NO=6
      IA=16
      IB=12
      IC=16
900      CALL VMULFF(PHIT2,D,L1,MO,NO,IA,IB,BCON,IC,IER)
      MO=5
      NO=1
      IB=6
905      CALL VMULFF(BCON,T,L1,MO,NO,IA,IB,3T,IC,IER)
      L1=1
      NO=6
      IA=1
910

```



```

          IC=1
          CALL VMULFF(TT,R,L1,MO,NO,IA,IB,RT1,IC,IER)
915      NO=1
          CALL VMULFF(RT1,T,L1,MO,NO,IA,IB,RT,IC,IER)
          RTI(1,1)=1./RT(1,1)
          L1=F5
          MO=1
920      IA=16
          IR=1
          IC=16
          CALL VMULFF(BT,RTI,L1,MO,NO,IA,IB,V1,IC,IER)
          DO 1083 J=1,F5
925      BTT(1,J)=BT(J,1)
          1187 CONTINUE
          L1=F5
          MO=1
          NO=F5
930      IA=16
          IR=1
          IC=16
          CALL VMULFF(V1,BTT,L1,MO,NO,IA,IB,VV,IC,IER)
          DO 1086 I=1,F5
935      DO 1085 J=1,F5
          QQ(I,J)=0.0
          IF(I.EQ.J)QQ(I,J)=1000.0
          1086 CONTINUE
          1087 CONTINUE
940      CALL MRIC(F5,ACN1,VV,QQ,S1,JLX1,TOL,IER)
          L1=1
          MO=1
          NO=F5
          IA=1
945      IR=1
          IC=1
          CALL VMULFF(RTI,BTT,L1,MO,NO,IA,IB,G01,IC,IER)
          MO=F5
          IR=16
950      CALL VMULFF(G01,S1,L1,MO,NO,IA,IB,G02,IC,IER)
          L1=6
          MO=1
          NO=F5
          IA=6
955      IR=1
          IC=5
          CALL VMULFF(T,G02,L1,MO,NO,IA,IB,G3,IC,IER)
          DO 1088 I=1,6
          DO 1087 J=1,6
960      BSP1(I,J)=BSP4(I,J)
          1087 CONTINUE
          1088 CONTINUE
          DO 1090 I=1,F5
          DO 1089 J=1,6
965      GCN1(I,J)=BCON(I,J)
          1089 CONTINUE
          1090 CONTINUE
          DO 1092 I=1,F5
          DO 1091 J=1,F5

```

AD-A094 766

AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCH00--ETC F/6 12/2
MODERN OPTIMAL CONTROL METHODS APPLIED IN ACTIVE CONTROL OF A T--ETC(U)
DEC 80 A M JANISZEWSKI

UNCLASSIFIED

AFIT/GS/AA/80D-2

NL

2 OF 2

AD-A094 766



END

DATE

FILED

3-81

DTIC

```
974      ULX(I,J)=ULX1(I,J)
      1191  CONTINUE
      1192  CONTINUE
      KK=KK+1
      DO 1200 I=7,9
975      D1(3,I)=D(I,3)
      120  CONTINUE
      N7=1
      GO TO 118
      1196  PRINT*,"SINGULARITY PROGRAM FAILED"
980      1101  STOP
      END
```

```

1      SUBROUTINE MRIC(N,A,S,Q,X,Z,TOL,IER)
      DIMENSION A(16),S(26),Q(16),X(16),Z(16)
      COMMON/MAIN1/NDIM,NDIM1,F(15)
      COMMON/MAIN2/TR(16)
5      COMMON/INOUT/NOT
      ANV=TOL*1.E-06
      NN=N*NDIM
      NM1=N-1
      IND=1
10     COUNT=0.
      IF(IER.EQ.1)COUNT=99.
      IF(IER.EQ.1)MR=N
      IF(IER.EQ.1)GO TO 100
      T1=-1.
15     300 CONTINUE
      IER=0
      COUNT=COUNT+1.
      DO 15 I=1,N
      DO 15 J=I,NN,NDIM
20     15 X(J)=-S(J)
      CALL INTEG(N,A,X,Z,T1)
      CALL FACTOR1(N,Z,X,MR)
      IER=1
      IF(MR.LT.0)GO TO 200
25     IER=0
      CALL GMINV(N,N,X,Z,MR,0)
      CALL TFR(TR,Z,N,N,1,2)
      CALL MMUL(Z,TR,N,N,N,X)
      DO 18 II=1,NN,NDIM1
30     I=II
      DO 17 J=II,NN,NDIM
      X(J)=(X(J)+X(I))/2.
      X(I)=X(J)
      I=I+1
35     17 CONTINUE
      18 CONTINUE
      100 CONTINUE
      DO 16 I=1,N
      16 TR(I)=-1.0
      TOL1=TOL/10.
40     MAXIT=40
      DO 40 IT=1,MAXIT
      IF(IER.EQ.1)GO TO 100
      CALL MMUL(S,X,N,N,N,F)
      CALL MMUL(X,F,N,N,N,Z)
45     DO 20 I=1,NN,NDIM
      II=I+NM1
      DO 20 J=I,II
      X(J)=A(J)-F(J)
      20 Z(J)=Z(J)+Q(J)
50     100 CONTINUE
      IER=0
      CALL MLINEQ(N,X,Z,X,TOL1,IER)
      IF(IER.NE.0)GO TO 200
      L=0
55     C1=0.0
      II=1
      DO 25 I=1,N

```

```

        IF(ABS(X(II))-TR(I)).LT.(ADV+TOL*X(II)) L=L+1
        TR(I)=X(II)
60      II=II+NDIM1
        25      C1=C1+TR(I)
        IF(ABS(C1).GT.1.E+20) GO TO 50
        IF(L.NE.N) GO TO 40
        CALL GMINV(N,N,Z,F,MR,0)
65      CALL MMUL(S,X,N,N,N,Z)
        DO 30 I=1,NN,NDIM
        II=I+NM1
        DO 30 J=I,II
        30      Z(J)=A(J)-Z(J)
        IF(MR.NE.N) WRITE(NOT,35) MR
        35      FORMAT(26H0RICCATI SOLN IS PSD--RANK13)
        GO TO 65
        40      CONTINUE
        WRITE(NOT,45) MAXIT
75      45      FORMAT(26H0RICCATI NON-CONVERGENT IN12,11H ITERATIONS)
        GO TO 60
        50      WRITE(NOT,55) IT,T1
        55      FORMAT(29H0RICCATI BLOW UP AT ITERATION12,12H INITIAL T=F10.5)
        60      IER=1
        55      RETURN
        200     IF(IND.EQ.2) GO TO 250
        IF(COUNT.GE.10.) RETURN
        T1=T1/(2.**COUNT)
        IND=2
85      GO TO 300
        250     T1=T1*(2.**COUNT)
        IND=1
        END

```

```

1      FUNCTION XNORM(N,A)
        DIMENSION A(16)
        COMMON/MAIN1/NDIM,NDIM1
        NN=4*NDIM
5      C1=0.
        TR=A(1)
        IF(N.EQ.1) GO TO 20
        I=2
        DO 10 II=NDIM1,NN,NDIM
        J=II
        DO 5 JJ=I,II,NDIM
        C1=C1+ABS(A(J)*A(JJ))
        J=J+1
        TR=TR+A(J)
15      I=I+1
        TR=TR/FLOAT(N)
        DO 15 II=1,NN,NDIM1
        15      C1=C1+(A(II)-TR)**2
        20      XNORM=ABS(TR)+SQRT(C1)
        RETURN
        END

```

```

1      SUBROUTINE MLINEQ(N,A,C,X,TOL,IER)
      DIMENSION A(16),C(16),X(16)
      COMMON/MAIN1/NDIM,NDIM1
      COMMON/MAIN3/F(16)
5      ADV=TOL*1.E-06
      DT=.5
      DT1=0.
      NN=N*NDIM
      DO 5 II=1,NN,NDIM1
10      DT1=DT1-A(II)
      DT1=DT1/N
      IF(DT1.GT.4.0)DT=DT*4.0/DT1
      II=1
      DO 20 I=1,N
15      DO 15 JJ=I,NN,NDIM
          X(JJ)=DT*A(JJ)
          X(II)=X(II)-.5
20      II=II+NDIM1
      CALL GMINV(N,N,X,F,MR,0)
25      IER=4
      IF(MR.NE.N)RETURN
      CALL MMUL(C,F,N,N,N,X)
      I=1
      DO 40 II=1,NN,NDIM
25      J=II
      IF(I.EQ.1)GO TO 30
      DO 25 JJ=I,II,NDIM
          C(J)=C(JJ)
30      J=J+1
      ID=J
      DO 35 JJ=II,NN,NDIM
          C(J)=CT*DOT(N,F(II),X(JJ))
35      J=J+1
      F(ID)=F(ID)+1.0
35      I=I+1
      DO 90 IT=1,20
          NEZ=0
          CALL MMUL(C,F,N,N,N,X)
          I=1
          II=1
          J=1
          GO TO 70
50      J=I
          DO 65 JJ=I,II,NDIM
          C(J)=C(JJ)
45      J=J+1
          ID=J
          DT1=C(J)
          DO 75 JJ=II,NN,NDIM
          C(J)=C(J)+DOT(N,F(II),X(JJ))
50      J=J+1
          J=J-1
          DO 80 JJ=II,J
          X(JJ)=F(JJ)
55      IF(ABS(C(ID)).GT.1.E+15)GO TO 95
          IF(ABS(C(ID)-DT1).LT.(ADV+TOL*ABS(C(ID))))NEZ=NEZ+1
          I=I+1

```

```

      II=II+NDIM
      IF(I.LE.N)GO TO 60
      IF(NE7.EQ.N)GO TO 150
      CALL MMUL(X,X,N,N,N,F)
      30 CONTINUE
      35 IER=1
      RETURN
65 150 CONTINUE
      NM1=N-1
      DO 155 I=1,NN,NDIM
      II=I+NM1
      DO 155 JJ=I,II
70 155 X(JJ)=C(JJ)
      IER=0
      RETURN
      END

```

```

1  SUBROUTINE FACTOR1(N,A,S,MR)
   DIMENSION A(16),S(16)
   COMMON/MAIN1/NDIM,NDIM1
   COMMON/INOUT/KOUT
5  TOL=1.E-06
   MR=0
   NN=N*NDIM
   TOL1=0.
   DO 1 I=1,NN,NDIM1
   R=ABS(A(I))
   1 IF(R.GT.TOL1)TOL1=R
   TOL1=TOL1*1.E-12
   II=1
   DO 50 I=1,N
   IM1=I-1
   15 DO 5 JJ=I,NN,NDIM
   5 S(JJ)=0.
   ID=II+IM1
   R=A(ID)-DOT(IM1,S(II),S(II))
   20 IF(ABS(R).LT.(TOL*A(ID)+TOL1))GO TO 50
   IF(R)15,50,20
   15 MR=-1
   WRITE(KOUT,1000)
   1000 FORMAT(37H0TRIED TO FACTOR AN INDEFINITE MATRIX)
   25 RETURN
   20 S(ID)=SQRT(R)
   MR=MR+1
   IF(I.EQ.N)RETURN
   L=II+NDIM
   30 DO 25 JJ=L,NN,NDIM
   IJ=JJ+IM1
   25 S(IJ)=(A(IJ)-DOT(IM1,S(II),S(JJ)))/S(ID)
   50 IT=II+NDIM
   RETURN
   35 END

```

```

1      SUBROUTINE INTEG(N,A,C,S,T)
      DIMENSION A(16),C(16),S(16)
      COMMON/MAIN1/NDIM,NDIM1,X(15)
      COMMON/MAIN2/COEF(16)
5      NN=N*NDIM
      NM1=N-1
      IND=1
      ANORM=XNORM(N,A)
      DT=T
10      5      IF(ANORM*ABS(DT).LE.0.5)GO TO 10
      DT=DT/2.
      IND=IND+1
      GO TO 5
10      10      DO 15 I=1,NN,NDIM
      J=I+NM1
15      DO 15 JJ=I,J
      S(JJ)=DT*C(JJ)
      T1=DT**2/2.
      DO 25 IT=3,15
20      CALL MMUL(A,C,N,N,N,X)
      DO 20 I=1,N
      II=(I-1)*NDIM
      DO 20 JJ=I,NN,NDIM
      II=II+1
25      C(JJ)=(X(JJ)+X(II))*T1
      20      S(JJ)=S(JJ)+C(JJ)
      25      T1=DT/FLOAT(IT)
      IF(IND.EQ.0)GO TO 100
      COEF(11)=1.0
30      DO 30 I=1,11
      II=11-I
      30      COEF(II)=DT*COEF(II+1)/FLOAT(I)
      II=1
      DO 40 I=1,NN,NDIM
      J=I+NM1
35      DO 35 JJ=I,J
      X(JJ)=A(JJ)*COEF(1)
      X(II)=X(II)+COEF(2)
      40      II=II+NDIM1
      DO 55 L=3,11
40      CALL MMUL(A,X,N,N,N,C)
      II=1
      T1=COEF(L)
      DO 55 I=1,NN,NDIM
45      J=I+NM1
      DO 55 JJ=I,J
      50      X(JJ)=C(JJ)
      55      II=II+NDIM1
      L=L+1
50      50      L=L+1
      CALL MMUL(X,S,N,N,N,C)
      II=1
      DO 90 I=1,N
      J=II
55      IF(I.EQ.1)GO TO 75
      DO 70 JJ=I,II,NDIM
      S(JJ)=S(J)

```



```

70      J=J+1
75      DO 85 JJ=I,N
        KK=JJ
60      DO 86 K=I,NN,NDIM
          S(J)=S(J)+C(K)*X(KK)
        KK=KK+NDIM
        J=J+NDIM
65      DO 87 JJ=1,NN,NDIM
        87      C(JJ)=X(JJ)
        90      II=II+NDIM
          IF(L.EQ.IND)GO TO 100
          CALL MMUL(C,C,N,N,N,X)
70      GO TO 60
        100     CONTINUE
          RETURN
          END

```

```

1      SUBROUTINE MMUL(X,Y,N1,N2,N3,7)
        DIMENSION X(16),Y(16),Z(16)
        COMMON/MAIN1/NDIM
        NEND3=NDIM*N3
        NEND2=NDIM*N2
        DO 1 I=1,N1
          DO 1 J=I,NEND3,NDIM
            Z(J)=0.0
            KK=J-I
            DO 1 K=I,NEND2,NDIM
              KK=KK+1
            1      Z(J)=Z(J)+X(K)*Y(KK)
          RETURN
          END

```

```

1      FUNCTION DOT(NR,A,B)
        DIMENSION A(16),B(16)
        DOT=0.
        DO 1 I=1,NR
          DOT=DOT+A(I)*B(I)
        1      RETURN
        END

```

```
1      SUBROUTINE TFR(X,A,N,M,K,I)
      DIMENSION X(16),A(16)
      COMMON/MAIN1/NDIM
      JS=(K-1)*NDIM*M
5      JEND=M*NDIM
      GO TO (10,30,50,70,90),I
      10  DO 20 II=1,N
          DO 20 JJ=II,JEND,NDIM
      20  X(JJ)=A(JJ+JS)
10      RETURN
      30  DO 40 II=1,N
          KK=(II-1)*NDIM
          DO 40 JJ=1,M
          LL=(JJ-1)*NDIM+II
15      40  X(KK+JJ)=A(LL+JS)
          RETURN
      50  KK=0
          DO 60 II=1,JEND,NDIM
          LL=II+N-1
20      DO 60 JJ=II,LL
          KK=KK+1
      60  X(KK)=A(JJ+JS)
          RETURN
      70  KK=M*N+1
25      DO 80 II=1,M
          LL=(M-II)*NDIM+1
          DO 80 IJ=1,N
          KK=KK-1
          JJ=LL+N-IJ
30      80  A(JJ+JS)=X(KK)
          RETURN
      90  SAVE=A(1)
          K=N
          DO 91 I=1,N
35      L=N
          DO 92 J=1,N
          IK=(K-1)*NDIM+K
          X(IK)=0.
          IF(L.EQ.K)X(IK)=A(L)
40      92  L=L-1
          91  K=K-1
          X(1)=SAVE
          RETURN
          END
```

```

1      SUBROUTINE GMINV(NR,NC,A,U,MR,MT)
      DIMENSION A(16),U(16)
      COMMON/MAIN1/NDIM,NDIM1,S(15)
      COMMON/INOUT/NOT
5      TOL=1.E-12
      MR=NC
      NPM1=NR-1
      TOL1=1.E-20
      JJ=1
10     DO 100 J=1,NC
      FAC=DOT(NR,A(JJ),A(JJ))
      JM1=J-1
      JRM=JJ+NRM1
      JCM=JJ+JM1
15     DO 20 I=JJ,JCM
20      U(I)=0.
      U(JCM)=1.0
      IF(J.EQ.1)GO TO 54
      KK=1
25     DO 30 K=1,JM1
      IF(S(K).EQ.1.0)GO TO 30
      TEMP=-DOT(NR,A(JJ),A(KK))
      CALL VADD(K,TEMP,U(JJ),U(KK))
30      KK=KK+NDIM
      DO 50 L=1,2
      KK=1
      DO 50 K=1,JM1
      IF(S(K).EQ.0.0)GO TO 50
      TEMP=-DOT(NR,A(JJ),A(KK))
35      CALL VADD(NR,TEMP,A(JJ),A(KK))
      CALL VADD(K,TEMP,U(JJ),U(KK))
50      KK=KK+NDIM
      TOL1=TOL*FAC
      FAC=DOT(NR,A(JJ),A(JJ))
35      54 IF(FAC.GT.TOL1)GO TO 70
      DO 55 I=JJ,JRM
      55 A(I)=0.
      S(J)=0.
      KK=1
40     DO 55 K=1,JM1
      IF(S(K).EQ.0.0)GO TO 65
      TEMP=-DOT(K,U(KK),U(JJ))
      CALL VADD(NR,TEMP,A(JJ),A(KK))
55      KK=KK+NDIM
45     FAC=DOT(J,U(JJ),U(JJ))
      MR=MR-1
      GO TO 75
70      S(J)=1.0
      KK=1
50     DO 72 K=1,JM1
      IF(S(K).EQ.1)GO TO 72
      TEMP=-DOT(NR,A(JJ),A(KK))
      CALL VADD(K,TEMP,U(JJ),U(KK))
72      KK=KK+NDIM
55     75 FAC=1./SORT(FAC)
      DO 80 I=JJ,JRM
      80 A(I)=A(I)*FAC

```

SUBROUTINE GMTNV

74/74 OPT=1

FTN 4.8+518

```

      DO 85 I=JJ,JCM
      95   U(I)=U(I)*FAC
      100  JJ=JJ+NDIM
      IF(MR.EQ.NR.OR.MR.EQ.NC)GO TO 120
      IF(MT.NE.0)WRITE(UNIT,110)NR,NC,MR
      110  FORMAT(I3,1HX,I2,8H M RANK,I2)
      120  NEND=NC*NDIM
      65   JJ=1
      DO 135 J=1,NC
      DO 125 I=1,NR
      II=I-J
      S(I)=0.
      70   DO 125 KK=JJ,NEND,NDIM
      125  S(I)=S(I)+A(11+KK)*U(KK)
      II=J
      DO 130 I=1,NR
      U(II)=S(I)
      75   130  II=II+NDIM
      135  JJ=JJ+NDIM
      RETURN
      END

```

SUBROUTINE VADD

74/74 OPT=1

FTN 4.8+518

```

1      SUBROUTINE VADD(N,C1,A,B)
      DIMENSION A(16),B(16)
      DO 1 I=1,N
      1   A(I)=A(I)+C1*B(I)
      5   RETURN
      END

```

SUBROUTINE XDOT

74/74 OPT=1

FTN 4.8+518

```

1      SUBROUTINE XDOT(TI,Y,YP)
      DIMENSION Y(20),YP(20)
      COMMON Z9(20,20)
      L1=20
      5   M0=20
      NO=1
      IA=20
      IB=20
      IC=20
      10  CALL VMULFF(Z9,Y,L1,M0,NO,IA,IB,YP,IC,IER)
      RETURN
      END

```

Vita

Alan Michael Janiszewski was born on January 24, 1951 in South Milwaukee, Wisconsin. He graduated from high school in South Milwaukee in 1969. After two years at the University of Wisconsin, he enlisted in the Air Force. During technical training he was selected to attend the United States Air Force Academy. He graduated with a regular commission and the degree of Bachelor of Science in Aeronautical Engineering in 1976. He was assigned to Warner Robins Air Logistics Center, Robins Air Force Base, Georgia. He served there as an Aeronautical Engineering Supervisor and a Programs and Plans Logistics Officer until his assignment to the AFIT School of Engineering in June 1979.

Permanent Address: 1601 Monroe Ave
South Milwaukee
Wisconsin

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GA/AA/80D-2	2. GOVT ACCESSION NO. AD A094766	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) MODERN OPTIMAL CONTROL METHODS APPLIED IN ACTIVE CONTROL OF A TETRAHEDRON		5. TYPE OF REPORT & PERIOD COVERED MS Thesis
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Alan M. Janiszewski Capt		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology AFIT-EN Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE December 1980
		13. NUMBER OF PAGES 98
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Approved for public release; IAW AFR 190-17 Fredric C. Lynch, Major, USAF Director of Public Affairs A.F.I.T. Director of Information 30 DEC 1980		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Linear system Feedback control Singular value decomposition Transformation matrix		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Modern optimal control methods are applied to a lumped mass model of a tetrahedron. The four unit masses of this model are interconnected by isotropic massless rods which are capable of axial deformation only (no bending). NASTRAN is employed in generating a normal modes approximation, while providing the mode shapes and frequencies for the resultant twelve modes. System control is achieved via collocated sensor/actuator pairs at three of the four masses. Initially, these sensors and actua-		

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

tors are given a prescribed line of operation. Pointing accuracy at the fourth mass is used as a figure of merit in determining the effectiveness of the controller. A prescribed bandwidth for line of sight error at 20 seconds is set as a goal for successful control.

The controller is developed using linear optimal techniques which produce feedback gains proportional to the state. The state is represented as modal amplitudes and velocities as determined by the sensors. The four higher frequencies modes are truncated to signify a simplifying order reduction step. State estimation is incorporated due to the non-availability of modal amplitudes and velocities. The feedback gains are established via steady state optimal regulator theory; this involves minimization of related quadratic performance indices. Control is applied with point force actuators. System response is examined in light of the effects of observation spillover and control spillover onto a specified number of suppressed modes. A comparison is obtained by complete elimination of the spillover effect. Using singular value decomposition, the spillover is first eliminated through judicious reorientation of one sensor/actuator pair. An attempt to control two modes and suppress six demonstrates the advantages of spillover elimination, but fails to satisfy the specified bandwidth for error.

Sensors are added to the model at the fourth mass and observation spillover is again eliminated. Reorientation of the initial sensor/actuator pairs is no longer applied. Line of sight response was improved over a case without sensor additions, but line of sight response was still inadequate. The truncated modes were added to the system with little degradation, verifying the acceptability of this truncation.

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

DATE
FILMED
-8